Assignment 4

● Graded

Group

Bikash Saha
Himanshu Karnatak
Valeti Lokesh
View or edit group

Total Points

100 / 100 pts

Question 1

Team Name

0 / 0 pts

✓ **+ 0 pts** Correct

+ 0 pts Incorrect / Level not cleared on the server

Question 2

Commands

5 / 5 pts

✓ **+ 5 pts** Correct: go --> dive --> dive --> back --> pull --> back --> back --> go --> wave --> back --> back --> thrnxxtzy --> read --> the_magic_of_wand --> c --> read

+ 0 pts Incorrect / Level not cleared on the server

Question 3

Cryptosystem

10 / 10 pts

✓ **+ 10 pts** ****Correct: 6-Round Data Encryption Standard (Des)

+ 0 pts Incorrect / Level not cleared on the server

− 2 pts Unnecessary story.

Question 4

Analysis

80 / 80 pts

✓ **+ 10 pts** 10 pts for Mentioning that the plaintext and ciphertext contain letters in the range f to u and the mapping of 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 wrong answer / error in code / Level not cleared on the server

+ 0 pts Plagiarism

+ 0 pts Late Submission

+ 0 pts Click here to replace this description.

Question 5

Password

5 / 5 pts

✓ **+ 5 pts** Correct

+ 0 pts Incorrect / Level not cleared on the server

Question 6

Code

0 / 0 pts

✓ **+ 0 pts** Correct

+ 0 pts Incorrect / Level not cleared on the server

Q1 Team Name

0 Points

Group Name

mod3

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 --> back --> back --> go --> 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-rounds of Data Encryption Standard (DES) block cipher with block size 64bit.

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.)

The hint provided by the spirit about the encryption method was either a 4-round or 6-round DES. We first decided to try breaking it as if it was a 6-round DES because it's generally more challenging to crack than the 4-round version. The 4-round DES is more easily broken, so we started with the stricter option. If that didn't work, we planned to see if it could be a 4-round DES and try breaking it that way. For this operation, we used Chosen-Plaintext attack and Differential cryptanalysis.

Determine the range of characters:

The spirit also hints at "two letters for one byte or something like that." So, we suspect that each byte contains two characters, which means each character is 4-bit in size. Therefore, a total of 16 characters can be present in the cipher text. So now we try to determine the range of the characters. We manually try out different plaintext and their corresponding ciphertext and determine the character's ranges within f and u. So we assign 0 to f, 1 to g and so on. At last, we assign 15 to u. The mapping of characters in binary form is provided below:
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

Differential Characteristic equation and generating Plaintext-Ciphertext pairs:

We have utilized two different differential characteristics equations to break the 6-round DES. These are 40 08 00 00 04 00 00 00 and 00 20 00 08 00 00 04 00, with probabilities of 1/16 each. There characteristics are the same as described in seminal work of Eli Bihem & Adi Shamir published in year 1990. Every characteristic allows us to find some key bits of sixth round key of DES. Combining the results from two characteristics allows us to find more key bits in lesser number of plaintext-ciphertext pairs. The master key can be found by brute force search over the remaining key bits.

For each characteristic equation, we generate a pair of plaintext. To generate the pair of plaintext for first characteristics, we apply inverse initial permutation on the characteristic 40 08 00 00 04 00 00 00 and obtain the XOR value between two pairs of plaintext as 00 00 80 10 00 00 40 00. Similarly, after applying inverse initial permutation on the second characteristic (00 20 00 08 00 00 04 00), we obtain the XOR between two plaintext as 00 00 08 01 00 10 00 00.

After obtaining the XOR value for each characteristics equation, we generate 2000 pairs of different plaintext using input_generator.py and obtained their corresponding ciphertext using output_generator.py. The plaintext pair for the first and second characteristics are stored in input_strings1.txt and input_strings2.txt, respectively, and store their corresponding ciphertext in output_strings1.txt and output_strings2.txt, respectively.

Breaking 6-Round DES:

After getting the ciphertext, we changed it into binary form based on the character mapping we had set up before. Next, we used DES_Breaking.ipynb to apply the reverse final permutation and got (L6 R6) and (L'6 R6') as the results from the 6th round of DES. Knowing that R5 equals L6, we used R5 and R5' to determine what comes from the Expansion box and the input XOR for the S-boxes during the 6th round.

We next determined the output of the permutation box by carrying out the calculation (L5 XOR L5') XOR (R6 XOR R6'), where (L5 XOR L5') is set to either 0x04000000 or 0x00000400, depending on the specific characteristic identified. We then applied an inverse permutation to this result to find the output XOR of the S-boxes for the 6th round. We introduced βi as (αi XOR k6,i) and βi' as (αi' XOR k6,i), where αi and αi' represent the outputs from the Expansion box for the respective ciphertexts, and k6 is the key for the 6th round. We calculated γi XOR γi' for each i ranging from 1 to 8.

The last round's Expansion box output splits as:
E(R5) = α1 α1.....α8 for and E(R5') = α1'α2'........α8' .
With βi = αi XOR k6,i and βi' = αi' XOR k6,i, where both αi and αi' are 6 bits in length, and k6 = k6,1, k6,2, ... k6,8.

Given this, we know αi, αi′, βi XOR βi', γi XOR γi′
where
where k ranges from 0 to 63, Si ranges from 1 to 8, γi XOR γi′ =S(βi) XOR (βi'), and E represents the expansion box.
We create a set such that:
Xi = {(β, β′) ∣ β XOR β′ = βi ⊕ βi′ and S(β) XOR S(β′) = γi XOR γi′}

Then, We obtained an 8x64 matrix to record how often each key k could potentially be the correct key for an a-Si box; we then evaluated the set Xi, identifying keys k that met the condition αi' XOR k = β for some pair (β, β') in Xi. Whenever a key k matched this criterion for a Si box, we updated its count in the matrix, increasing key_matrix[i][k] accordingly.

This detailed process helped us identify potential keys, resulting in findings for the specific characteristic 40 08 00 00 04 00 00 00. The key frequency table is shown below:

S-box	Max	Mean	Key	Diff
S1	121	65	45	56
S2	302	77	59	225
S3	115	64	37	51
S4	99	66	7	33
S5	165	72	13	93
S6	306	73	55	233
S7	172	72	9	100
S8	170	68	58	102

And for the second characteristic (00 20 00 08 00 00 04 00), we get the key frequency as below:

S-box	Max	Mean	Key	Diff
S1	173	69	45	104
S2	173	69	59	104
S3	131	67	37	64
S4	312	83	7	229
S5	168	69	13	99
S6	304	74	55	230
S7	112	65	9	47
S8	111	65	58	46

Find the original key:

Observing these two tables the 6th round key become:
101101111011100101000111001101110111001001111010

Then, we utilize the key scheduling algorithm and find the master key. The obtained master key looks like:
011X11X00101111001X11011100X0101001X10111010X01X1101X011

Here X means the bit value is unknown for that position. At this position we have 8 unknown bit values. Then, we iterate over all 2^8 possible values to obtain the full master key.

Obtained Master Key:
01101110010111100111101110000101001110111010001111010011

After we got the master key, we generated all possible round keys. The rounds key are shown below:

Round 1   111111000100111100000111001011011001011010110110
Round 2   011011110011111101100010100110100111001111101000
Round 3   111010101111010011101101011100001111101100100101

Round 4	110110011100011101011010111100100010110010111010
Round 5	011001001101101110111011111011010011101100011011
Round 6	101101111011100101000111001101110111001001111010

Password Recover:
After finding the whole key, we type the word password in the game server to obtain the ciphertext of the original password. The cyphertext was ‘fnmgpspgfrnrmrrsfruqksihhhrfqolh’.
Given that each character corresponds to 4 bits, the 32-character ciphertext translates into a 128-bit sequence or two DES ciphertext blocks. Utilizing the master key, we decrypted this sequence through a 6-round DES decryption process we obtained [113, 117, 115, 106, 106, 99, 99, 115, 103, 97, 48, 48, 48, 48, 48, 48] in decimal value as a decrypted password. Then, we find the ASCII value of these numbers and retrieve the plaintext as "qusjjccsga000000," which includes additional zeroes as padding. Upon removing these zeroes, the plaintext "qusjjccsga" was used as the final password to advance to the next level in the game.

Q5 Password

5 Points

What was the password used to clear this level?

qusjjccsga

Q6 Code

0 Points

Please add your code here. It is MANDATORY.

DES_Final_Code.zip Download

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