Assignment 4

● Graded

Group

KUNAL SAINI
KUSHAL MAHESHWARI
SHARATH KUMAR V

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

Resolved **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

↻ Regrade Request **Submitted on: Apr 28**

We have mentioned that after getting the 42 bits by breaking the S boxes, we applied brute force to get the rest 14 bits(that is, try all 2^14 possibilities) in the analysis part. Still, 10 marks have been deducted for this.

updated.

Reviewed on: Apr 29

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

hardwired

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)

enter -> jump ->jump-> back -> pull -> back -> back -> enter -> wave -> back -> back -> thrnxxtzy -> read -> the_magic_of_wand -> c -> read -> password -> c -> stswgvftfd-> c

Q3 Cryptosystem

10 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.)

we reached on the screen which gave the information that the 4 - level or 6 - level DES has been used for this level.

Since 4 level DES would have been too easy to crack , we decided to try 6 level DES first to decrypt the cipher text .

The screen also mentioned that " two letters for one byte or something like that" which means that 2 letters are coded using 8 bits which further implies that each letter is encoded using
4 bits . So , the domain of the plaintext that can be encrypted using the above mentioned encoding can clearly contain atmost 16 letters.

On whispering password , we got the text
"kgrrhjhlnuqiqtjiumjnmiupjjpljhqf"
After few trail and error plaintexts we got to know that the ciphertext contains only letters from f to u. So we mapped them to 4-bit binary numbers.
The most natural mapping from alphabets to 4-bit binary would then be M : f -> 0000 , g -> 0001 , h -> 0010 , ... , u-> 1111 .

Note also that we need a lot of plaintext-cyphertext pairs in order to extract the 56 bit key used in the DES. We found out that on typing anything except the word "password" , we got a 16 letter word as output . Clearly , the output was the encryption of the text we type. So this provided a way to get the DES Encryption corresponding to any 16 letter plain text .

Differential Cryptanalysis:

We Used the technique of Differential Cryptanalysis to break the 6-round DES. The first component of Differential Cryptanalysis is to get the Characteristic Equation. Following the analysis provided in the article(https://medium.com/lotus-fruit/breaking-des-using-differential-cryptanalysis-958e8118ff41) for 6-round DES , we observe that using two 3

round characteristics , we can easily crack the DES . So we use the two 3 round Characteristics as mentioned in the article.
□1 = 4008000004000000 = 1/4
and
□2 = 0020000800000400 = 1/4

The next step is to generate plaintext-cyphertext pairs to be used for Breaking the DES. Note that it is a chosen plaintext attack , so we use pairs of plaintext that have XOR value RIP(4008000004000000)corresponding to the first characteristic and pairs of plaintext that have XOR value RIP(0020000800000400) corresponding to the second characteristic where RIP(x) denotes the application of inverse of Initial permutation IP[] to 64 bit input x . This has to be done because the input to the DES is first passed through the permutation IP[] before feeding it to the actual DES encryption scheme.

We have generated 7000 pairs of such plaintexts each for Characteristic 1 and 2 . These plaintexts are stored in plain_ch1.txt and plain_ch2.txt that are generated using io_generation.py. The pair of plaintext are generated such that their xored values equals corresponding characteristics. To get the corresponding Ciphertext for these plaintexts, we have used the bash script execute.sh that uses the ssh command prompt to get the encryption corresponding to the plaintexts in files plaintext1.txt and plaintext2.txt based on the provided characteristic number(either 1 or 2). The generated file is post-processed using gen_out.py to extract the ciphertexts. The corresponding Ciphertexts get stored in the ciphertext1.txt and ciphertext2.txt .

Extracting the Key:

At this stage we have the pairs of plaintexts (with the required XOR values) and their corresponding encryptions with us.

Let (ck, ck+1) denote the binary form of ciphertexts corresponding to the input plaintexts (pk, pk+1) where pk, pk+1 are such that pk ⊕ pk+1 is the desired XOR. Note that to get the binary form from the aplhabetical form, we use the one-one mapping \textbf{m} as defined above.

Now for each pair (ck, ck+1) for k ∈ [1 -7000], we do the following :
We pass ck through the inverse of final permutation i.e RFP . This gives us the actual output of the DES encryption algorithm , call it ak. The left and right

halves of αk give us the values L6 and R6 respectively . Since R5 = L6 , we pass the value R5 = L6 through the expansion function to obtain αk = E(R5) . Repeat the same for ck+1 to obtain αk′ = E(R5′) .
Now Since βk ⊕ βk′ = αk ⊕ αk′where βk, βk′ denote input to the S-box (after taking XOR with the key) , we have obtained the value of XORed input to the S-box cooresponding to each pair (ck, ck+1) for k ∈ [1 -7000]. Now to obtain the XORed value of the output of S-boxes corresponding to each (ck, ck+1) , do the following :

For each pair (ck, ck+1) for k ∈ [1 - 7000], we do the following :

Calculate R6 , R6' corresponding to ck and ck′ as mentioned above .
Now also note that we have the XORED value of L5 (with a certain probability and corresponding to certain S-boxes only) , call this value l5 . Now calculate Pk = l5 ⊕ R6 ⊕ R6′ . Now pass this value Pk through the inverse_permutation , the obtained value is the Xored output γk ⊕ γk′ which is correct corresponding to the relevant S-boxes. Note that the relevant S-boxes. in case of Characteristic 1 are S2, S5, S6, .. and in case of Characteristic 2 are S1, S2, S4, S5, S6 only . This is because these are the S-Boxes that have Zero input XOR (and hence Zero output Zero with Probability 1) in the forth round of the DES as mentioned in (https://medium.com/lotus-fruit/breaking-des-using-differential-cryptanalysis-958e8118ff41)

So the above analysis gives us pairwise XORs of the Output of relevant S-boxes for each pair (ck, ck+1) for k ∈ [1 - 7000].
Now to obtain the Keys corresponding to the relevant S-boxes , we perform the Frequency Analyis (as mentioned in the lectures) .
For each pair of plain-text , ciphertext we calculte the set Xi as defined below
Xi = { (βi, βi′) ∣ β ⊕ β′ = βi ⊕ βi′ and Si(β) ⊕ Si(β′) = γi ⊕ γi′ } .
Now for each such pair (β, β′) , calculate the 6-bit key corresponding to the S-box □i Ki = β ⊕ αi where αi are the output bits of the expansion function corresponding to S-box □i . To maintain the frequency count of different keys corresponding to diiferent S-Boxes , we maintain a 2-D array keys[8][64] and do keys[i][(int)(Ki)] ++ for each key Ki obtained corresponding to S-box i . (Here (int)(Ki) refers to the integer value corresponding to the six bit key).

Now , Since the above calculation makes sense only for S-boxes S2, S5, S6, .. in case of Characteristic 1 , we find the keys that occurs the most number of times corresponding to these S-Boxes in the analysis for Characteristic 1 and similarly we get the most occuring keys for S1, S2, S4, S5, S6 in case of analysis of Characteristic 2 . These most frequently occuring keys are the required correct sub-key corresponding to the S-boxes. Note that since the S-boxes S2, S5, S6 are common in both the Characteristics , we get the key values corresponding to only 5+5-3 = 7 S-Boxes , which means we get 7*6 = 42 bits out the total 56 bits.

Now , in order to calculate rest of the 56-42 = 14 bits , we use brute force technique wherein we enumerate all the 2^14 possibilities corresponding to the unknown bits and get the particular key that satisfies the plain-text to cipher-text mapping.
The Final key obtained by the analysis then is :
Final Key : 0110111001011110011110111000011100001100001101111011011

Now , Using this Key in the DES algorithm , we got the plain-text corresponding to the given cypher-text "kgrrhjhlnuqiqtjiumjnmiupjjpljhqf" by first splitting it into two halves of 16 alphabets each (16 alphabets -> 64 bits) and then concatenating the corresponding binary plain-texts to get the plain-text (in binary) :

011100110111010001110011011101110110011101110110011001100111010001
10011001100100001100000011000000110000001100000011000000110000

Now , we tried to convert the above binary plain-text into alphabets using the reverse mapping M as defined above and got "mimjmimmlmmlllmjllljififififififif" as the plain-text .
 But on typing this in the propmt we got nothing interesting!!.
So , we thought to convert the above 128 bit binary into alphabets using the standard Ascii mapping where 8-bits are used to represent each character. On doing so, we got
stswgvftfd000000 as the answer.
 On Entering this again nothing interesting happens !!, so we remove the Zeros and type "stswgvftfd" which clears the level 4 !!.

Q5 Password

5 Points

What was the password used to clear this level?

stswgvftfd

Q6 Code

0 Points

Please add your code here. It is MANDATORY.

▾ **break_6_round_des.zip** Download

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