Q1 Team Name

0 Points

The_Kryptonians

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)

$$go \rightarrow dive \rightarrow dive \rightarrow back \rightarrow dive \rightarrow pull \rightarrow back \rightarrow back \rightarrow back \rightarrow enter \rightarrow wave \rightarrow back \rightarrow back \rightarrow thrnxxtzy \rightarrow read \rightarrow 134721542097659029845273957 \rightarrow c \rightarrow read \rightarrow password$$

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.)

Reaching the encrypted password:

We observed light and felt a blast of air coming from the right side.So we went right and saw a lake. We dived into the lake and noticed a wand in the lake. We tried to pull the stuck wand but came out of the lake due to lack of breath. Then we dived again and pulled out the magic wand.Then we went back to the first screen using command 'back'. Then we went back again to level 3. We entered into a hole where we free the spirit by waving the wand. We then went back to the main chamber in level 3 and used commands 'thrxnntzy' -> 'read -> 134721542097659029845273957 to clear the level and reach the first screen of level 4.

Now we use the command 'read' to read with the help of spirit what is written on the glass panel. The spirit informed us that DES can be 4 round or 6-round. It also mentioned that 2 letter for 1 byte has been used which means 1 letter is of 4 bits.

When whispered word 'password' we get the ciphertext or our encrypted password - "qmiqiejqrhpfnqdhgngpkmmkfpdddjem" . On entering any other command we received the corresponding ciphertext.

Assuming cryptosystem used to be 6-round DES we approached in the following way :

1. Generating Input-Output pairs through chosen plai

Since we have the probability of 1/256, thus we intended to generate 20 x 256 $\approx$ 5000 plaintexts. It was observed that the ciphertext consisted of 16 letters namely { d,.....,s}. Thus we created a one-to-one mapping and generated the plaintext in a similar format where d corresponds to '0000', e to '0001', ... , and finally s to '1111'.

The 2 characteristics we used are: **4008000004000000** and **0020000800000400**, and for each of them we generated 5000 input pairs stored in **plaintext.txt** and **plaintext1.txt**

respectively. It was ensured that the XORed value of the pairs of plaintext generated like the 1st and 2nd text, the 3rd and 4th text is equal to the reverse initial permutation of the corresponding characteristic. We can find 30 bits out of 48 of K6. However, 3 S-Boxes (S2, S5 and S6) are common, hence we get $\mathbf{42}$ out of the 56 bit key set, and brute-force to get the remaining $\mathbf{14}$ bits.

We generated input plaintexts pairs using function $\mathbf{generate_input}()$and stored them in plaintext.txt and plaintext1.txt. We created two additional input text files (int.txt and int1.txt). Then we extracted the outputs of these input files from the server (172.27.26.188)
using terminal command “ssh students@172.27.26.188 < in.txt > out.txt”(Similarly for int1.txt) . We then cleaned these output files and stored the remaining output in ciphertext.txt and ciphertext1.txt files respectively.

2. Breaking 6 round DES using differential cryptanalys

We used our mapping of letters $\mathbf{d} \ldots\ldots \mathbf{u}$ to convert ciphertext to corresponding binary form of 64 bits. We applied reverse final permutation to these converted ciphertexts through our function rev_final_permutation and from that we get values of L6 and R6 and L6’ and R6’ from the first and second halves . As $L_{i+1} = R_i$, we pass R5 and R5’ through our apply_expansion function to get our α and α’. Since we know XOR of output of Expansion is equal to XOR of input of SBox. We took pairwise XOR {$(b_i \oplus b_{i+1})$,$(b_{i+2} \oplus b_{i+3})$ ... } of output obtained from the $\mathbf{expansion\ function}$ to get input for the SBox.

As mentioned in DES- differential-cryptanalysis.pdf , for 1st characteristic our input XORs for five Sboxes (S2, S5, S6, S7, S8) are zero therefore their output XORs are zero. Similarly for 2nd characteristic input XORs for five Sboxes (S1, S2, S4, S5 and S6) are zero so their output XORs are zero. As $R_4 = L_3 \oplus f(R_3)$, here f(x) function represents the fiestal output of x. We don’t know the exact value or XORed value of $L_5 (L_5 = R_4)$. But we do know the output XORs of some Sboxes are zero. We can use these XOR outputs to find 6-bit key values corresponding to those Sboxes of subkey k6 (k6 is the subkey for the round 6).
We apply inverse permutation to R6 by passing it through $\mathbf{apply\_inv\_permutation}$ and compute output XORed valued of each Sbox at round 6 by taking pairwise ciphertexts and xoring them.

Now we initialise a 2D array keys[8][64] with zeroes . This array is used to store count of 6-bits keys of each Sbox. As taught in lectures for each xored input Sbox we find pairs (β, β') such that X_i = {(β, β') | $\beta \oplus \beta$' = $\beta_i, \oplus\beta_i$' and $S_i(\beta) \oplus S_i(\beta')$= $\gamma \oplus \gamma$' }. Now we take any of these pairs (β, β') and XOR them with output of the expansion function to get a 6-bit key corresponding to that SBox. We convert binary form of key to integer and increase the count of this 6-bit key for that Sbox. We are using this to calculate frequencies of a particular 6-bit key for each sbox. As taught in lectures we find the most frequent 6-bit key corresponding to Sboxes {S2, S5, S6, S7. S8 } for 1st characteristics and Sboxes {S1, S2, S4, S5, S6} for 2nd characteristics. Now we have 30 bits for each characteristics. "#" has been used to represent unknown bits.

Since Sboxes S2 , S5 and S6 are common in both characteristics their computed key values should be same in both characteristics. After merging 6-bits for every Sbox we recovered 42 bits. Following is the table for most frequent 6-bit key corresponding to each Sbox in five Sboxes for each characteristics.

For characteristic : 40 08 00 00 04 00 00 00

$S-Box$	$6-bitkey$	$Frequency$
2	111011	40
5	000101	25
6	010101	43
7	000000	20
8	111000	25

For characteristic : 00 20 00 08 00 00 04 00

$S-Box$	$6-bitkey$	$Frequency$
1	101101	24
2	111011	24
4	000111	43
5	000101	26
6	010101	41

The partially obtained 56 bit key (14 unknown bits and 42 known bits) is: ['#', '1', '1', '#', '#', '1', '#', '#', '0', '1', '0', '1', '1', '#', '1', '0', '0', '#', '#', '1', '1', '#', '1', '1', '1', '1', '0', '0', '#', '0', '1', '0', '1', '0', '0', '0', '#', '1', '0', '0', '0', '0', '0', '0', '0', '#', '0', '1', '#', '0', '1', '0', '1', '#', '0', '1', '1']

Brute-forcing our way to get the remaining 14 bits, we obtain the final 56-bit key:

01101110010111100111101110000101000110000000001

From the above 56-bit key, the 48-bit keys subsets for each of the 6 rounds are:

Round1 key:101101111011100101000111000101010101000000
Round2 key:011001001101101110111011010010010001101100
Round3 key:110110011100011101011010111000100000110000
Round4 key:111010101111010011101101010000001011100100
Round5 key:011011100111111011000100000001001010001111
Round6 key:111111000100111100000111001011001000010000

The key are then used to decrypt the password, and we get the decrypted password in binary format as:

011100010111100001110011011011110110011101111000
011001100111000100110000001100000001100000001100

Note: Even when input consisted of characters other than {d,...s}, ciphertext was obtained. Thus we can have any character in plaintext. On converting it using ASCII code, the obtained decrypted password was: **qxsogxbtfq000000**. However, it wasn't accepted until we tried removing the trailing zeroes.

Thus, the final password is:

qxsogxbtfq

No files uploaded

Q5 Password

5 Points