Q1 Team Name

0 Points

flash

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/jump->dive/jump->back->dive->pull->back->back->go->wave->back->thrnxxtzy->read->134721542097659029845273957->read->passoword

For the next logins we have used read and passoword for getting the encrypted password

Q3 CryptoSystem

5 Points

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

6 round DES(Data encryption standard)

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.)

In the magical screen, we were presented with the following hints.
"The code used for this is a 4-round DES, so it should be easy for you!! Er wait ... maybe it is a 6-round DES ... sorry, my memory has blurred after so many years. But I am sure you can break even 6-round DES easily. A 10-round DES is a different matter, but this one surely is not 10-round ... (long pause) ... at least that is what I remember". So it is clear from the hint that it is not a 10 round DES, it can be either 4 or 6 round des. We decided to first try with 6 round des and if it doesn't work we can modify it to 4 round des.

We obtained cypthertext as 'ipphsdlmgffnophhjdkggfokoiehddfj' (after using read and then password command) that need to be decryted using des and then used as a command to clear level 4.

we made use of chosen plaintext attack to break 6-round DES. In this type of attack for cryptanalysis the attacker generates a number of plaintexts, gets the sender to encrypt them anyway and then uses the obtained pairs of plaintexts and ciphertexts for finding the key used for encryption.

Following functions are used in the DES algorith

IP(M) : This is applied on the plaintext M which is to be encrypted.
E(M): Expnad 32-bits of text M to 48-bits.
P(M): This step permutes the 32-bit input M.
S : there are 8 S-boxes. Each S-boxes has 6-bit input and a 4-bit output.
PC1: key permutation that maps 64 bits of keys to 56 bits and removes the parity bits.
shift- shift that is permormed on the key obtained as output of PC1
PC2: key permutation that maps 56 bits of shift's output to 48 bits.
IP_INV(M): This is applied after all 6 rounds of DES are done on message M.

Procedure followed to break the 6 round DES:

We performed differential cryptanalysis using two 3-round characteristics and used chosen-plaintext attack for cryptanalysis of 6-round DES. The characteristics used are 40080000 04000000$(characterstics1)$ and 00200008 00000400 $(characterstics2)$.

4 bits are used to represent one letter as it was mentioned that one byte consists of 2 letters.

#Using 4 bits we can only represent 16 letters.Thus, we tried several plaintexts and analysed the corresponding ciphertexts to identify which 16 letters are used in the game for this level.

#After doing the analysis on the ciphertexts we inferred that alphabets from d to s are used in the game for this level. Therefore, we mapped letters from d to s with 0 to 15 respectively:
{d : 0000,e : 0001, f: 0010, g : 0011, h : 0100, i : 0101, j : 0110, k 0111, l: 1000, m : 1001, n: 1010, o : 1011, p: 1100, q : 1101, r : 1110, s : 1111}

- The input and output size of one DES block is 64 bits i.e. 8 bytes (block size) which means 16 $(i.e., 64/4 = 16)$ letters. Thus, we decided to generate the plaintexts of size 16 letters.

Step 1:Generating plaintexts:

The differential characteristic 4008000004000000 with probability 1/16 and 0020000800000400 with probability 1/16 are used. We generated 1000 pairs of plaintexts and ciphertexts corresponding to each characteristic to break 6-round DES. The first 1000 plaintext pairs are generated such that their XOR was 0000801000004000 which is obtained by applying inverse initial permutation on the characteristic 400800000000000 and another 1000 plaintext pairs such that their XOR was 0000080100100000 which is obtained by applying inverse initial permutation on the characteristic 0020000800000400 These inputs are stored in plaintexts1.txt and plaintexts2. txt respectively. The code for generation of plaintext pairs is in generate_inputs_flash.ipynb,

Step 2: Obtaining Ciphertexts corresponding to

We automated the collection of ciphertexts corresponding to the plaintexts from server, we used Python's pexpect to establish connection to the server using valid credentials. We used plain_to_cipher1.py to generate the ciphertexts corresponding to plaintexts stored in plaintexts1.txt and plain_to _cipher2.py to generate the ciphertexts corresponding to plaintexts stored in plaintexts2.These ciphertexts are stored in ciphertexts1.txt and ciphertexts2.txt respectively.

Step 3: Finding the key bits of round key K6:

We carried out steps from a to d for the ciphertexts obtained corresponding to each of the two characteristics.

#a : We used the mapping of letters defined above to convert the obtained ciphertext to binary and then, we used flash_cryptanalysis.ipynb to apply reverse final permutation on these binary ciphertexts to get (L_6, R_6) and ($L_6^{'}$,$R_6^{'}$), which is output of the 6^{th} round of DES. We know that, $R_5 = L_6$.Thus, using the values R_5 and $R_5^{'}$, we computed output of Expansion box and input XOR of S-boxes for 6^{th} round.

#b : For the first characteristic mentioned above, L_5 =04000000 and for the second characteristic L_5=00000400. We found output of permutation box by performing $L_5 \oplus (R_6 \oplus R_6^{'})$, then we applied inverse permutation on this value to obtain output XOR of S-boxes for 6^{th} round.

#c : Let $E(R_5) = \alpha_1\alpha_2.....\alpha_8$ and $E(R_5^{'}) = \alpha_1^{'}\alpha_2^{'}.....\alpha_8^{'}$ and $\beta_i = \alpha_i \oplus k_{6,i}$ and $\beta_i^{'} = \alpha_i^{'} \oplus k_{6,i}$, where $|\alpha_i| = 6 = |\alpha_i^{'}|$ and $k_6 = k_{6,1}k_{6,2}.....k_{6,8}$. At this point, we know $\alpha_i, \alpha_i^{'}, \beta_i \oplus \beta_i^{'}$ and $\gamma_i \oplus \gamma_i^{'}$. We created a 8 * 64 key matrix to store the number of times a key k ∈ [1,64], satisfies the possibility of being a key to S_i box, where i ∈[1,8].

#d : We computed the set $X_i = (\beta, \beta_i^{'}) | \beta \oplus \beta^{'} = \beta_i \oplus \beta_i^{'}$ and $S(\beta) = \gamma_i \oplus \gamma_i^{'}$. Then we found the key k , such that $\alpha_i \oplus k = \beta$ and $(\beta, \beta^{'}) \in X_i$ for some $\beta^{'}$. For all the keys k which

satisfied this condition for S_i box, we incremented their count in the key matrix i.e., key_matrix[i][k] was incremented

#After performing the above analysis to find the keys, we obtained
the following results for characteristic 4008000004000000:

S-box Max Mean Key Diff
S1 132 71 61 61
S2 326 80 59 246
S3 125 66 37 59
S4 106 67 7 39
S5 167 74 41 93
S6 319 71 49 248
S7 179 71 23 108
S8 179 69 54 110

For the above characteristic, in round 4, XOR will be zero for S2, S5, S6, S7 and S8. Therefore, in round 6 these S-boxes will give the corresponding key bits of K_6. It can also be observed that a significant difference is their in the maximum key frequency and mean key frequency for these S-boxes which further assures of these key values being correct. We proceeded by taking the key bits for S2, S5, S6, S7 and S8 boxes as 59, 41, 49, 23 and 54 respectively.

#The above analysis gave the following results for characteristic 0020000800000400:

S-box Max Mean Key Diff
S1 156 67 61 89
S2 156 68 59 88
S3 130 68 37 62
S4 308 80 7 228
S5 185 70 41 115
S6 304 77 49 227
S7 122 67 23 55
S8 109 67 54 42

For the above characteristic, in round 4, XOR will be zero for S1, S2, S4, S5 and S6. Thus, in round 6 these S-boxes will give the corresponding key bits of K_6 Also, it can be observed that a significant difference is seen in the maximum key frequency

and mean key frequency for these S-boxes. We proceeded by taking the key bits for S1, S2, S4, S5 and S6 boxes as 61, 59, 7, 41 and 49 respectively.

Both the characteristics have S2, S5 and S6 as common S-boxes and we obtained same key values for these three S-boxes which further verified that our computations so far are correct.

Therefore, we proceeded by taking key values for S1, S2, S4, S5, S6, S7 and S8 as 61, 59, 7, 41, 49, 23 and 54 for round key K_6 Thus, at this point we know 42 bits of the 56 bit key.

Step 4: Find the Actual Key from 42 known bits:

#Next, we applied key scheduling algorithm to obtain the actual positions of these known 42 bits in the 56 bit key and obtained the following result:

Masterkey=X11XX1XX01011X100XX11X11101X0110001X11111001X10X0111X001, here X denotes unknown bits.

#At this point we have 14 unknown bits and for these 14 unknown
bits of DES key, we iterate through all 2^{14} possible permutations of the key to find the correct
key. We took plaintext= ddddddddddddddd and the corresponding ciphertext= dilkjieheoqkiedi and performed 6 round DES encryption. The key which encrypts this plaintext to produce the correct
ciphertext is the final key. From this step, we obtained the following key which satisfied the above results:

#Actual 56 Bit key =
01101110010111100111101110100110001011111001010001110001

#After obtaining the 56 bit key, we generated the 48 bit round key for each round.

ROUND KEY IN BINARY
Round 1 111111000100111110000111000101111100011010001110
Round 2 011011110011111101100010000100000011111100011110

Round 3 11101010111111001110110101111101001100001011010
Round 4 11011001111001110101101001100001011010001110101 1
Round 5 01100100110111111011011001001101011100000011111
Round 6 11110111101110010100011110100111000101011111011 0

Step 5: Decrypting password(ciphertext):

#The ciphertext corresponding to our password is 'ipphsdlmgffnophhjdkggfokoiehddfj' and thus to obtain the password we performed decryption on this ciphertext.

#This ciphertext has 32 letters. Since, each letter is represented by 4 bits, so this is 128 bit string, that is, 2 blocks of DES ciphertext. As per our mapping this is {92,196,240,137,50,42,188,68,96,115,50,183,181,20,0,38}

#Now that we have our key, we perform decryption on this ciphertext by considering 16 letters(=64 bits) at a time using decrypt. cpp, which uses decryption function of DES implementation for 6 rounds.

#The plaintext obtained is - srsseigthy000000. We removed the zeroes as they might have been used for padding.

#We entered the plaintext ‘srsseigthy’ in the game and were directed to the next level. This is the code to clear the 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?

srsseigthy

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.

Flash_Assignment4.zip Download

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

Assignment 4

GRADED

GROUP

Nimit Jain
Ravi Shankar Das
Prashant Mishra
View or edit group

TOTAL POINTS

71.5 / 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 — **-27.5** / 0 pts