Q1 Team Name

0 Points

Pasta_Sandwich

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, dive, back, pull, back, back, go, wave, back, back, thrnxxtzy, read, 134721542097659029845273957, c, read, 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.)

After using the commands mentioned above and collecting the magic wand, We return to the first screen of Level-4 and type the command read. The Spirit here hints towards DES. It says it can be DES-4 rounds which is very easy or DES-6 rounds. The spirit strikes out the possibility of DES-10 rounds by mentioning 'This one is surely not 10 rounds.....'
We tried DES-4 rounds but that approach didn't work so we decided to go with DES-6 rounds. The spirit mentioned that " there was something funny about the way the text appeared, two letters for one byte or something like that.' As DES has a block size of 8 Bytes, we will be having 16 letters in a block. We gave a few random 16 letter inputs and observed their outputs. All the letters in the output were in the range of [d,s]. 16 letters can be represented using 4 bits, so we mapped the letters d to s to 0-15.
The mapping is as follows:
{'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'}

In order to break the 6-round DES, we use chosen plain-text attack. We get the plain texts encrypted from the server and use them for finding the keys.

1. We performed differential cryptanalysis using 2 three-round characteristics 40080000 04000000 and 00200008 00000400.
- Generating Plain-text Pairs:
The differential characteristic 40080000 04000000 and 00200008 00000400 both with probability 0.0625 were used.
- We generated 1000 random plaintext pairs such that the XOR of each pair is 0000801000004000. 0000801000004000 is the inverse permutation of the characteristic 40080000 04000000. We store the generated pairs in plaintext1.txt
- We generated another 1000 random plaintext pairs such that the XOR of each pair is 0000080100100000. 0000080100100000 is the inverse permutation of the characteristic 00200008 00000400. We store the generated pairs in plaintext2.txt.
- The code for generating the plaintext pairs, is available at Input_Generator.ipynb

2. Extracting the Cipher-texts:
- We used the python script (cipher_generator.py) to extract the ciphertext for the corresponding plaintext.
- Ciphertexts corresponding to plaintext1.text and plaintext2.text are stored in ciphertext1.txt and ciphertext2.txt respectively.

3. Finding the 6th round key bits:
- Converted the obtained ciphertexts to binary using the available mapping.
- Performed reverse permutation on the converted ciphertexts($L_6 R_6$ and $L'_6 R'_6$) to obtain the output of 6th round DES. The output of the expansion box and the input XOR of 6th round S-boxes can be computed using R_5 and R'_5 as R_5 is known($R_5 = L_6$)
- The output of the permutation box can be evaluated by performing L_5 xor R_6 xor R'_6, applying inverse permutation to the same gives the xor output of the 6th round s-boxes.
- As mentioned in lecture slides, 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 $|\alpha_i| = |\alpha_i'| = 6$, $k_6 = k_{6,1}k_{6,2}...k_{6,8}$.
- Therefore, we know values of $\alpha_i, \alpha_i', \beta_i \oplus \beta_i'$ and $\gamma_i \oplus \gamma_i'$. Then we made 8x64 key matrix which stored the number of times a key between 1 to 64, satisfies the possibility of being a key to sbox S_i for each sbox.
- After this, we calculated the set $X_i = (\beta, \beta')|\beta \oplus \beta'$. i.e $X_i = \beta_i \oplus \beta_i'$ and $S(\beta) \oplus S(\beta') = \gamma \oplus \gamma'$. Then, we were able to find the key k, such that $\alpha_i \oplus k = \beta$ and ($\beta, \beta') \in X_i$ for some β'.
- From all the keys we obtained which keys have satisfied the condition for sbox S_i, key matrix's count was incremented.

The steps mentioned above are performed for both characteristics.

- In round 4, the XOR will be zero for S2, S5, S6, S7 and S8 for the characteristic 4008000004000000. Therefore, the round 6 outputs of these s-boxes will be the corresponding key bits of K_6 . The key bits for S2, S5, S6, S7 and S8 are 51, 26, 62, 13, 62 respectively. The frequency of the key occurrence can be found in cryptAnalysis.ipynb.
- For the other characteristic, round 4 XOR will be zero for S1, S2, S4, S5 and S6. Therefore, the round 6 outputs of these s-boxes will be the corresponding key bits of K_6. The key bits for S1, S2, S4, S5 and S6 are 45,51,7,26 and 62 respectively. The frequency of the key occurrences can again be found in cryptAnalysis.ipynb.
- Both the characteristics have S2, S5 and S6 as common s-boxes and the keys retrieved for these s-boxes are also same for both the characteristics.
Therefore, the final key values of the s-boxes S1, S2, S4, S5, S6, S7 and S8 are 45, 51, 7 , 26, 62, 13 and 62 respectively.

4. Finding the Master key:
-In the previous step, 42 bits of the key were retrieved. We applied the key-scheduling algorithm to the 42 bits to obtain the master key.
-We obtain the Master key as X11XX1XX01011X100XX11X11000X1001111X0111010X10X1101X011, where X is the unknown bit.
-There are 14 unknown bits. We use brute force to determine the same. We iterate over all the 2^14 combinations.
-We take the plaintext: ‘dddddddddddddddd’ (00000000) and the corresponding ciphertext: oemhmklisgomdonl to retrieve the remaining bits of the key.
-The code is available at bruteforce.py

Retrieved Key: 01101110010111100111101100001001111001111010110111011011

5. Decrypting the password:
- Our encrypted password is ‘gljhllppmilrmrnqhsglqeqqijfqhlfq’. This ciphertext contains 32 characters. In order to decrypt this ciphertext, we first find the 2 blocks of the ciphertext. According to our mapping, the ciphertext is {56,100,136,204,149,142,158,173, 79, 56, 209, 221, 86 ,45 ,72 ,45}
The code for the same can be found in conversion.py
-We perform decryption using the retrieved key taking 64 bits at a time. The decryption is performed using decryption.cpp which is an implementation of the decryption algorithm of 6 round DES.
-Finally, we get our plaintext as ‘putxjteife000000’ . We removed the extra zeroes from the end, which were supposedly added as padding bits.
Finally, we get the the plaintext as ‘’putxjteife” and enter the same as password.

No files uploaded

Q5 Password

5 Points

What was the password used to clear this level?

putxjteife

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.

▾ Assignment-4.zip Download

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

Select a question.

Group Members | Submission History | Next Question ›