CS641: The Great Caves

DedSec

Guntas Singh Brar
180274

Divyanshu Bhardwaj
180253

Devanshu Gupta
180233

March 1, 2020

Chapter 4 :The Spirit

Reaching the Cipher Text

- We enter the fourth chamber and hear loud rumbling sound. We see a door to the left and a large opening on the right. We "read" the glass panel but there is nothing written on it. So we think of exploring the entrance on the right.
- We "go" through a rocky pathway to reach a lake with a big waterfall.We "dive" into the lake and lose breath. So we "dive" again and we see a rod looking like a wand at the bottom. We try to pull the wand and we lose breath again. We "dive" in third time and pull out the wand. We try to explore further but find nothing in there.
- Then we remember the message on the previous door:

  "BREAKER OF THIS CODE WILL BE BLESSED BY THE SQUEAKY SPIRIT RESIDING IN THE HOLE. GO AHEAD AND FIND AWAY OF BREAKING THE SPELL ON HIM CAST BY THE EVIL JAFFAR. THE SPIRIT OF THE CAVE MAN IS ALWAYS WITH YOU. FIND THE MAGIC WAND THAT WILL LET YOU OUT OF THE CAVES. IT WOULD MAKE YOU A MAGICIAN NO LESS THAN JAFFAR!"
- So we go back to the previous chamber to the squeaky spirit in the hole and "wave" the wand in front of the hole. The spirit gets freed. Now we go back to the fourth chamber and as we try to "read" the panel, we hear the spirit whispering in our ears:

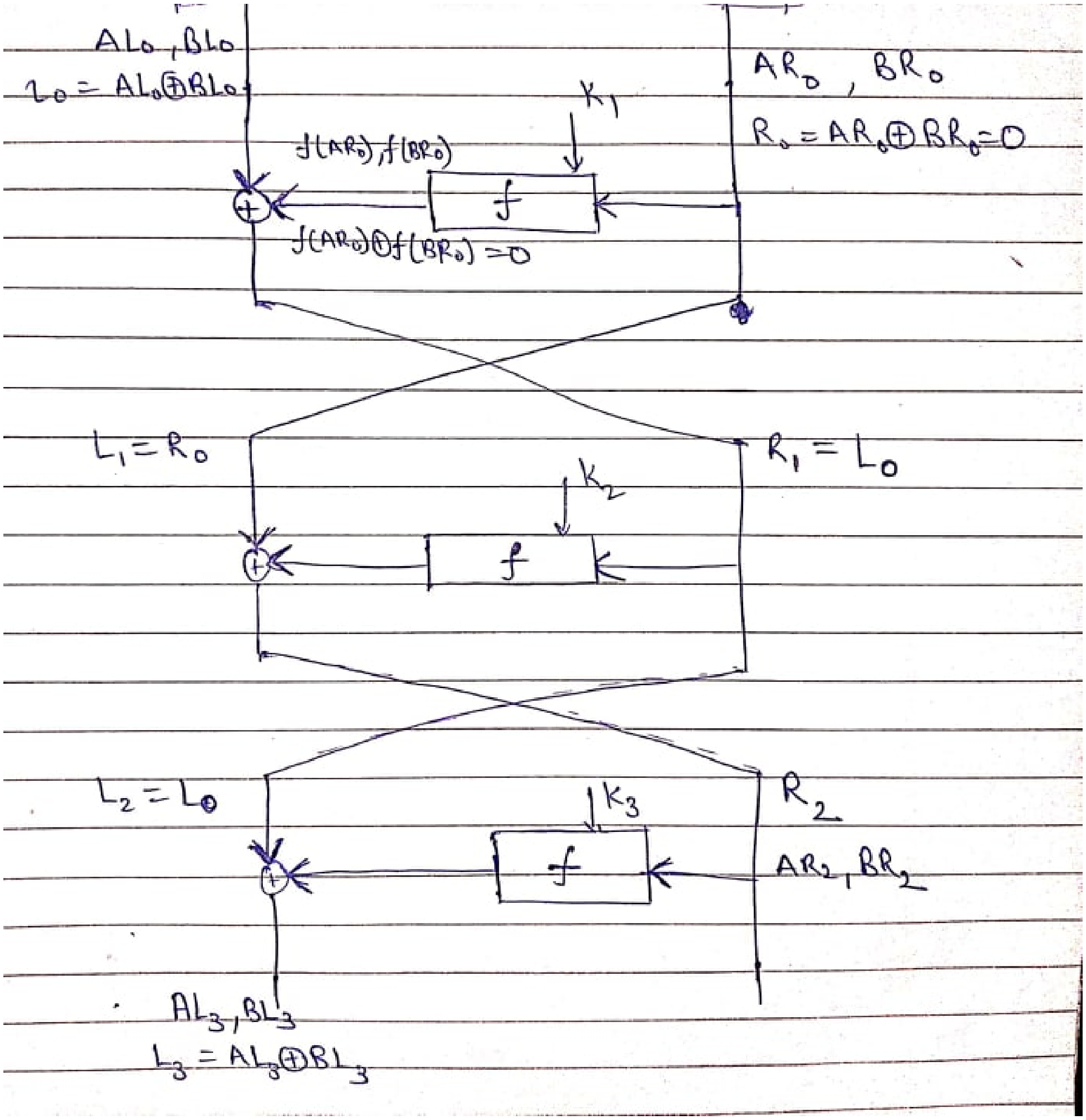
  "This is a magical screen. You can whisper something close to the screen and the corresponding coded text would appear on it after a while. So go ahead and try to break the code! The code used for this is a 4-round DES, so it should be easy for you!! Er wait ... maybe it is a 6-round DES ... sorry, my 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. One thing that I surely remember is that you can see the coded password by whispering 'password'. There was something funny about how the text appears, two letters for one byte or something like that. I do not recall more than that. I am sure you can figure it out though ..."

Cracking The Cipher Text

- So as we know it is DES Encryption, we try differential cryptanalysis.But before that we notice something strange about the cipher text. As it is DES encryption, the cipher text must either be in binary or hexadecimal. For 8 character long plain text(64bit), we get cipher text 16 character long. The binary output of the encryption surely will be 64bit long(and the cipher text has to be the ASCII translation of binary into characters, it should've been 8 character long)

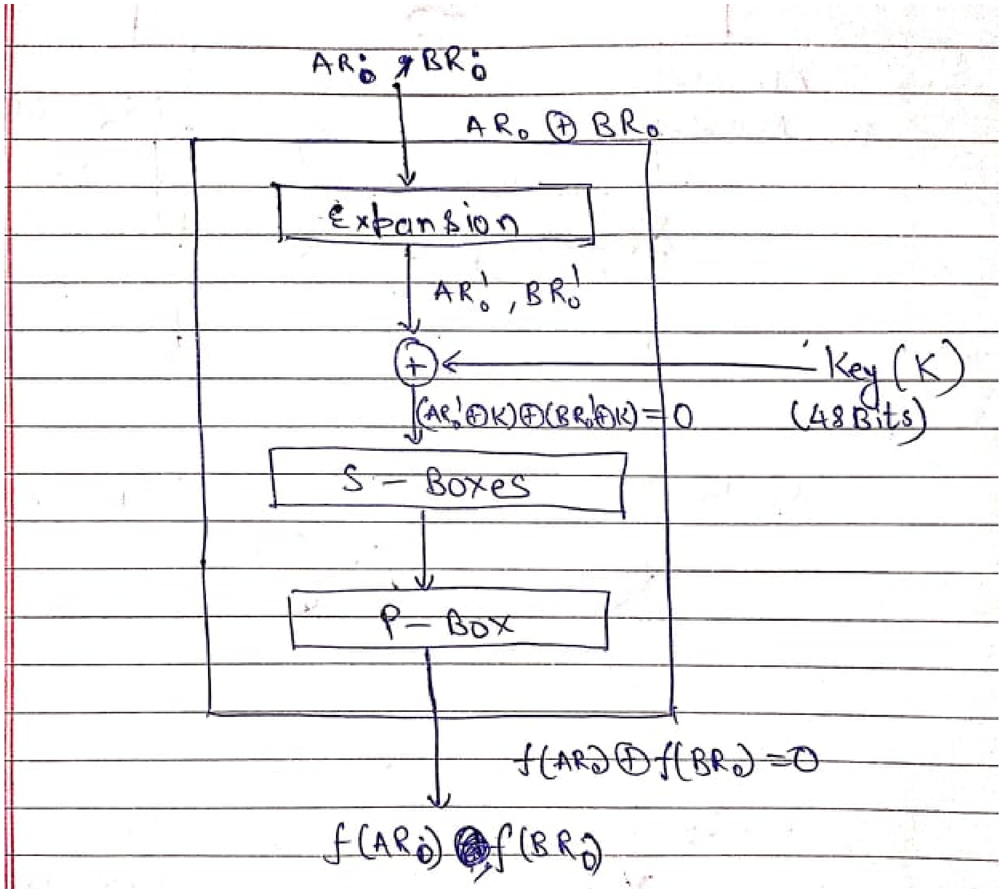
[*]

- One character for every 4 bits(16bit long cipher-text) means that it must be hexadecimal. But we don't see any numbers in the cipher-text for any plain-text. Then we notice that for any plain-text, only 16 characters(f-u) occur in the plain-text. We think of the possibility that the 16 characters of the hexadecimal notation(0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f) might have been mapped to the 16 characters u to f.
- Now according to the spirit it can be a 4-round or 6-round DES. But observing the spirit, it
- So we try chosen plain-text differential cryptanalysis attack for 3-Round DES as we were told in the lectures that it is a 3-round DES encryption.
- Now we first analyse the encryption a bit and we choose plain texts in pairs, such that after the initial permutation(ip in constants.py file) the xor of the right halves is 0 or in other words the right halves are identical. As shown below:



- Here, AR_i and BR_i denote the right whereas AL_i and BL_i denote the left halves of the input of each round. R_i and L_i are their respective xors.

- Below we have shown why $f(AR_o) \oplus f(BR_o) = 0$.(xor being zero means that the bits are identical. So if input bits in Expansion, S-box and P-box are same, output bits will also be the same):

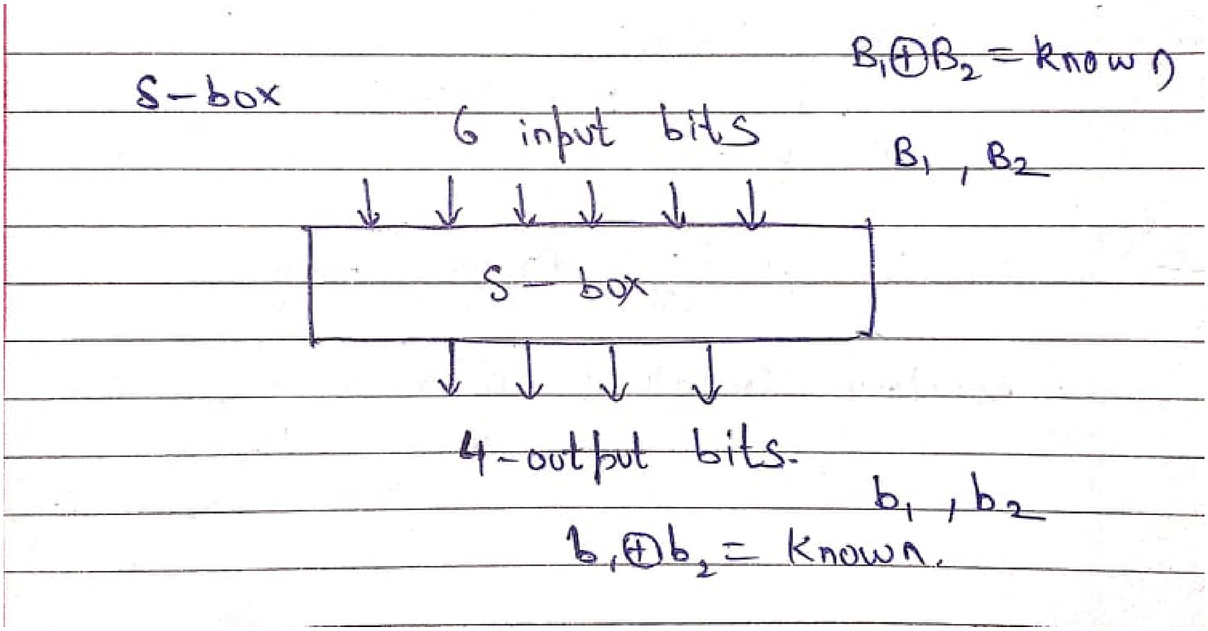


$$(AR_0' \oplus K) \oplus (BR_0' \oplus K) = (AR_0' \oplus BR_0') \oplus (K \oplus K)$$

$$= (AR_0' \oplus BR_0') = 0$$

- Now finally we know AR_2, BR_2 and R_2. Also we know AL_3, BL_3 and L_3. We know L_2 so we know the xor of the two outputs of the feistel function in the third round. Also we know both the inputs(AR_2 and BR_2) and their xor (R_2).Now we proceed on finding the 48bit key for the third round.

- So now we know the xor of input bits and the xor of output bits in the s-box. So, we have total 2^6 possibilities for each s-box(Choosing B_1 automatically fixes B_2 as we know the xor) and total 2^9 possibilities(8 s-boxes). So now we look for the cases in which the xor of output bits equals the xor that we have. Now that we know the two inputs that are xor-ed with the key and we know the corresponding two outputs(from above algorithm) we can determine the 48 bit key.



- Now once we run the above algorithm on 5-6 pairs of plain text. We can easily narrow down to one possibility of the 48bit key for last round. Once we get there, we brute-force on the remaining 8 bits to get the 56bit Key and hence we crack the encryption.(The still remaining 8 bits are parity bits that can be calculated but not required here.)
- On running the above algorithm we get the 48 bit key for the third round as:

"000000001111000100111111111110110100000101001000"

- And the 56 bit key as:

"10100101011001001011000110001101011000001001100010100101"

- We decode the password cipher text to get the password for the next level:

"rushmnfshsfnqfjgnrjokfokiifknumq"

Attachments

The following files are attached:

- **48bitkey.py** :Gives the 48bit key of the 3rd round
- **56bitkey.py** :Gives 56bit key through brute-force
- **DES.py** :Implementation of 3-round DES
- **constants.py** : Constants required for encryption