CS641: Level 1
Team: TrojanHorse

Members:
Ayush Kumar, 180174
Rishav Kumar, 180612
Nilay Majorwar, 180483

February 29, 2020

Reaching the instructions:

Reaching the instructions was tricky in this level:

1. Speak `go` at the first screen.
2. `dive` into the lake.
3. Diving more, we run out of breath and die. So instead, go `back` to the surface.
4. `dive` back into the lake, and `pick` the wand.
5. Go `back` to the surface, and go back to level 3.
6. `wave` the wand in front of the two holes. The spirit is freed, and follows us now.
7. Go to level 4, and `read` the glass panel to get the instructions.

Instructions on glass panel:

"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 ..."

Breaking the alphabet-binary encoding:

It was told in the classroom that it's a 3-round DES. We observed that on entering a plaintext of length 1-16 i.e. we get ciphertext of length 16. Thus, 16 characters has to correspond to one full block. Since one full block is 8-byte long, 2 characters have to correspond to one byte.

Also notice that the output contains of only 16 letters - from `f` to `u`. Now, 16 letters can be represented with 4 bits, so by concatenating the 4-bit formats of two characters, we get the corresponding byte.

In short,

```
plaintext[i]= (inputtext[2*i]-'f')*16 + (inputtext[2*i+1]-'f')
```

Getting the round three key:

We will launch a differential cryptanalysis attack to extract the key of the third round. We generated 10 differential pairs (see `third.py`), and applied the inverse of the initial permutation `ip` to get the texts we need to enter into the game (see `cio.py`). We note the obtained ciphertext pairs to get the full differential text pairs. Now we can launch the attack.

Now consider the case of a particular differential pair. As taught in class, we find the xor of outputs of permutation box of round 3 (using xor of L_2). From this, we find the output xor of S-boxes. Also, we know the individual outputs of the expansion box of round 3, and can obtain the xor of the inputs of the S-boxes.

So we know the input xor and output xor of each S-box. We will get a maximum of 64 input pair possibilities for each S-box, and thus 64 possibilities for that block of the key. Doing this for each differential pair, the intersection of all these possibility sets reduces to one value. Concatenating these key blocks, we get the full key for the third round. (see `third.py`)

Getting the first two round keys:

Once we get the third round key, the problem reduces to a 2-round DES. We generate 6-7 plaintexts with corresponding ciphertexts (see `breakdes.py`). For a 2-round DES, we know the text of the middle stage too (due to the L-R swap). So we have the general problem of solving a 1-round DES with known plaintexts and ciphertexts.

We can easily get the output of S-boxes, and will get 4 possibilities for the input of each S-box, which is the same as the output of key xor. We also know the input of key xor (which is just the expansion of R_0). So, we get 4 possibilities for the corresponding block of the key. Doing this for each plaintext-ciphertext, we reduce each set of possibilities to one, and concatenating these key blocks, we get the round key (see `breakdes.py`).

Using this algorithm, we get the round keys for both the first round and second round. Thus, we obtain all the keys for the three-round DES.

```
KEY 1:  110011101111001010111101011010011111000000101000
KEY 2:  111111011100010111100111000110101100111000101001
KEY 3:  111100111100101110011011000110100101110100010000
```

Getting the decrypted password:

Now, we just need to decrypt the encrypted password (`jfkifgqoorgrrnqnmssiiqislkftkurl`). Since the encrypted password has 32 characters, the decrypted password consists of two blocks. Splitting the encrypted password into blocks, we decrypt each block to get `rqmjnkisnpqirgpl` and `prrjfosfspqomhik`. Concatenating the two blocks, we get `rqmjnkisnpqirgplprrjfosfspqomhik`, which on entering in the game, takes us to the next level.

– – – – – – – – – – – – – – – – – – – –