CS641 - Modern CryptoGraphy
Assignment 4

Shashvat Singham
(200922)

On entering Chapter 4, initially we find a screen with nothing useful written on it. Navigating through the passage we reach a lake. Diving in the lake, we see a wand stuck at the bottom of the lake. We come to the surface for breath, dive again and pick the wand. Then we went to the rat which we found in a hole in Chapter 3 and wave the wand before it. The rat turns into a soul, says that it will guide us and leaves. We again went to the beginning of Chapter 4 and read the screen which initially had gibberish. The screen now shows some useful message along with a message from the soul we just freed.

The Cipher used in this level is 3-round DES. We implemented the Differential Cryptanalysis of 3-DES to find the keys used in the encryption. Here, we assumed that it is a standard 3-round DES, with 48-bit round keys generated from a 56-bit master key through key-scheduling.The steps involved in breaking DES is detailed as below:

1 Encoding String

The message said, "two letters for one byte". This means that each letter is encoded in 4 bits meaning that the valid input/outputs have only 16 distinct letters. On analysing the different outputs for different inputs, we found that only letters f-u appear in the outputs. Now we assumed that f is encoded as 0000, g as 0001 ..... and u as 1111. Further the input is padded to next multiple of block-size. We didn't know the padding scheme in use and hence provided inputs of length 16 characters (i.e. 8 bytes when encoded) to avoid padding.

2 Differential Cryptanalysis

In the description below we denote input after applying initial permutation by [L0, R0] and corresponding 3-round output before initial permutation reverse by [L3, R3]. Since IP and IP-INV are known they can easily be reversed. In the description below we omit them. However, they have properly been handled in the code we submit.

First of all, we computed the differential table i.e. given the input and output XOR values of an S-Box, what can be the possible inputs of the S-Box. We did this for all the 8 S-Boxes. We then take pairs of chosen plaintexts such that R0 in both inputs in the pair are same. We then compute the input and output XOR values to the S-Boxes as follows:

$$input\_xor = E(L3) \oplus E(L3')$$

$$output\_xor = P^{-1}((R3 \oplus L0) \oplus (R3' \oplus L0'))$$

where,

E = Expansion function
P = Permutation function
Unprimed and Primed variables are first and second input/output in the pair respectively.

Using the input and output XORs, we find the possible values of third round keys. We do this for multiple pairs and take the intersection of possible key values for each pair until only one key value is possible. Just 13 pairs of Input/Output was required to find the 3rd round key.

After we found the 3rd round key, we use the key scheduling in reverse to map 3rd round key bits to the master key bits. Thus, we get 48 bits of master key. The remaining 8 bits are easily brute-forced as there are only 256 possibilities. Out of the 256 possibilities, only 1 master key value satisfies all the 13 pairs that we observed. Thus, we get the master key.

3 Password

On whispering password near the screen we get the encrypted password. Using the key that we extracted, the password can easily be decrypted.

4 Code

The code decrypt.py has the decryption code. It can be run as

python3 decrypt.py

It prints out the master key found and the decrypted password. The master key is actually 64-bit, 8 of which are parity bits which are not used in the algorithm. We simply set them to -1 (equivalent to Dont Care)(key is represented as a list of bits).