ASSIGNMENT 4

CS-641

TEAM- Bullshot

Members –

Aman Tayal(180074)

Rohit Ranjan(180629)

Rishabh Kothary(180608)

For this Assignment we had to break 3 Round DES(Data Encryption Standard).We used the following commands to reach the ciphertext go -> dive -> dive -> back -> pull -> back -> back -> go -> wave ->back -> thrnxxtzy -> read -> xtf_ijwib_ttt -> c -> read -> password It was a chosen plaintext attack where we needed to decipher the following ciphertext:

“mnqfjfkpgrrsjgpqfkgkltggrtgrtiro”

We were given the hint that 2 letters constitute a byte but we had to first figure out how exactly. Our first key observation was that all the output was in the range of letters “f to u”, which consists of 16 elements. Thus our guess was that the input from the range “f to u” can be uniquely determined and if we give f the value 0 and go serially and give u, the value 15, then each letter can be represented by 4 bits and thus 2 letters constitute a byte. To prove our claim that f = 0, we first find the ciphertext corresponding to any plaintext (not a multiple of 16) say “gh” and the ciphertext turned out to be:

“gmqqulgqjrqklpul”

And the ciphertext to the plaintext “ghf” turned out to be:

“gmqqulgqjrqklpul”

Which is exactly the same thus proving our claim that f corresponds to 0. We observed that the ciphertext was of 32 characters implying it was of 128 bits, thus there are 2 blocks of data of 64 bits which are encrypted. In order to decipher the given ciphertext we needed to find the key first. The key of DES is of 64 bits where 8 bits are parity bits, thus effectively key of DES is of 56 bits. Brute-forcing the key was not a viable of option as the total number of possible keys are 2^{56} which is roughly equal to 10^{16}. We used differential cryptanalysis to break 3 round DES. We took input pairs such that after the initial permutation, their XOR was of the form (L’,0), both parts represent the left and the right half of the plaintext. Then after first round it transforms to (0,L’) and after second round to some (L’,F(L’)) and after third round to (F(L’),F(F(L’))+L’).

By moving throught the F function we were able to find the input XOR and output XOR of the S boxes. We took 10 pairs of data plaintext, cipher text pairs solutions. The actual key would be a solution to every pair. Thus after finding all possible keys using the s-box input and output XOR pair we the found the following data:

sbox : 1 key: 51,freq: 10,confidence: 100.000000

sbox : 2 key: 23,freq: 10,confidence: 100.000000

sbox : 3 key: 12,freq: 10,confidence: 100.000000

sbox : 4 key: 4,freq: 10,confidence: 100.000000

sbox : 5 key: 15,freq: 10,confidence: 100.000000

```
sbox : 6 key: 1,freq: 10,confidence: 100.000000
sbox : 7 key: 22,freq: 10,confidence: 100.000000
sbox : 8 key: 21,freq: 10,confidence: 100.000000
```

From the above analysis we got the 48 bits of Key. Then backtracing the key generation algorithm we expanded the key to its 56 and finally to its 64 bit form where we find the rest of the 8 bits of the key using brute forcing where we need to do $2^8 = 256$ computations which is fairly reasonable.

The key found (64-bit form) turned out to be:
1110101100001010010001001111010110000111000010100110111101010000

Using the above found key we decrypted the given ciphertext which turned out to be:
1100110110101011000001010001100101011001101001110011100001001011001010111110110101101110000011011001110101111101001111001001010 0

Which according to the mapping we found earlier converts to:

“rspqfkgokopminjqhqtsltfsosmsiroj”

And that is how we cracked it. The code used to crack it is attached in the folder.