Report MitE[2]

Irad Nuriel
(324220458/irad_juixanf_2a3cabbd55227e164b5e76bd109cb112edee3d2570dc1e2e5)

March 8, 2021

1) Attack:

   The attack I’ve implemented is a partial meet in the end attack, I found that a change in nibbles: 2,7,8,12,13,14 doesn’t change the 10th nibble of the ciphertext after 4 rounds of TC07.(and, I only inserted a possible key to the hash table only if it made a match on the 10th nibble with 4 of the 16 plaintext-ciphertext I had(as I don’t has an insane amount of more than 1TB of RAM).

2) Optimizations/data structures:

   In my attack, I used a hash set(std::unordered_set) because of the O(n) space complexity and O(1) access time(and it was the structure my competitive programmers friends suggested I should use), and as my attack only finds the keys which match on the 10th nibble, I needed to BF the remaining 16 bits of the key.
   I also combined the MC and SR in the round function, and moved all the implementation into a cell representation.

3) Building the program:

   Run make in the folder (see the Makefile for more information).

4) Problems:

   It was pretty hard for me to find a good mask for the key, until I realized that I could check for each nibble of the key on what nibble of the ciphertext it doesn’t affect after 4 rounds of TC07, and then just if two nibbles don’t affect the ith nibble of the cipher text after 4 rounds, then the combination of them will not affect the ith nibble after 4 rounds either(most of the times,

sometimes it isn’t, but I can easily check if that the case or not).
I also had a hard time debugging as each run would take me ages(2-3 hours actually)

5) Extra:
    a) I first found a 16 bit mask(instead of 24) that a change in them doesn't change the 7th nibble of the ciphertext after 4 rounds of TC07 by hand.
    b) and I found 4 masks of 24 bits that a change in them doesn’t change a certain nibble in the ciphertext after 4 rounds of TC07.