Report Brute Force[1]

Irad Nuriel
(324220458/irad_juixanf_2a3cabbd55227e164b5e76bd109cb112edee3d2570dc1e2e5)

March 8, 2021

1) Attack:

   The attack I've implemented is a very basic Brute Force attack(running in 2^{32} computations and 2 memory).

2) Optimizations:

   The optimization I've done is to move the code into a native language(C), and activate the compiler optimization flags -O3 -march=native(led to about 100 times speedup).

   Another optimization I did was to move the Sbox loop from iterating over i from 0 to 15 and with jumps of 1, and shifting by i*4(requires to calculate i*4 2 times in each iteration), to a loop over i from 0 to 64 with jumps of 4, basically replacing i*4 with i, which made the program run faster by a bit(less than 1 time speedup, but moving into cell representation would make the process slower as doing the L function in cell representation would be impossible(the <<<15 is not nibble aligned and left rotation isn't feasible in cell representation) or slower(we would have to move between representations and it will be slow)).

3) Building the program:

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

4) Problems:

   Measuring the runtime of the program was pretty hard for me as I haven't done such a thing for a long time.

5) Extra:

a) I checked the path of trying to implement the cipher with a cell representation instead of state representation, and it happened to be about 5 times slower.
b) When compiling the code with make, you will get 2 ELF files, one for the brute forcing called bruteforce, and one I used for time profiling called timeProfiler.