Hints

- Without the Sbox, all operations preserve bit-wise linearity. In other words, if a block of plaintext has 128 bits $b_1, b_2, \cdots b_{128}$, then every single ciphertext bit has the form

$$c_0 + c_1 b_1 + c_2 b_2 + \cdots + c_{128} b_{128}$$

- It's possible to find $c_1, c_2, \cdots, c_{128}$ without knowing the key, or you can recover all c_i from the plaintext-ciphertext pairs.

- If you have issues solving linear equations in booleans in python, try python package `galois` (https://github.com/mhostetter/galois):

```
import numpy as np
import galois

GF2 = galois.GF(2)  # create the binary field
M = GF2([[0, 1], [1, 0]])  # build a 2x2 matrix
v = GF2([1, 0])  # build a vector of length 2
solution = np.linalg.solve(M, v)  # solve linear equation Mx = v
first_number = int(solution[0])  # get the first entry of the solution
```