**Mission 6 Name:**

Use the Sage code demonstrated in class to attack an SDES system. Do all of your work in the Mission 6

CoCalc Assignment files. (They will be collected as well.)

**Part 1:** Use differential cryptanalysis to attack SDES (3 rounds).

You encrypt the plaintext P=[0,0,0,0,1,1,1,1,1,0,1,1] and get the ciphertext C=[0,1,0,0,1,1,1,1,1,1,0,0]. In particular L3 = [0,1,0,0,1,1] and R3=[1,1,1,1,0,0].

In order to attack the system, you also use several values for a second plaintext, P\*. The values of the plaintext, along with the ciphertext and the corresponding values of the xor of the inputs to the sboxes, (E(L3)⊕E(L\*3)) and the outputs ((R3⊕R3\*)⊕(L0⊕L0\*)) are given. Determine the value of K3.

**First alternate plaintext:** P\* = [1,0,0,1,1,0,1,1,1,0,1,1]

(E(L3)⊕E(L\*3)) = [1,1,1,0,1,0,0,1] ((R3⊕R\*3)⊕(L0⊕L0\*)) = [1,0,0,1,0,1] Possible values of input to Sbox 1 (From L0):

[0, 1, 1, 1]

[1, 0, 0, 1]

Possible values of input to Sbox 2 (From L0):

[0, 0, 0, 1]

[0, 0, 1, 1]

[0, 1, 0, 0]

[1, 0, 0, 0]

[1, 0, 1, 0]

[1, 1, 0, 1]

**Second alternate plaintext:** P\* = [0,1,0,1,1,1,1,1,1,0,1,1]

(E(L3)⊕E(L\*3)) = [1,0,0,0,0,0,1,1] ((R3⊕R\*3)⊕(L0⊕L0\*)) = [0,1,1,0,1,1] Possible values of input to Sbox 1 (From L0):

[0, 1, 0, 0]

[0, 1, 0, 1]

[0, 1, 1, 1]

[1, 1, 0, 0]

[1, 1, 0, 1]

[1, 1, 1, 1]

Possible values of input to Sbox 2 (From L0):

[1, 0, 0, 1]

[1, 0, 1, 0]

[1, 1, 0, 1]

[1, 1, 1, 0]

**Third alternate plaintext:** P\* = [0,1,1,0,1,1,1,1,1,0,1,1]

(E(L3)⊕E(L\*3)) = [1,1,0,1,0,1,0,1] ((R3⊕R\*3)⊕(L0⊕L0\*)) = [1,1,0,1,1,1]

Possible values of input to Sbox 1 (From L0):

[0, 1, 1, 1]

[1, 0, 1, 0]

Possible values of input to Sbox 2 (From L0):

[1, 1, 0, 1]

[0, 0, 0, 1]

[0, 1, 0, 0]

[1, 0, 0, 0]

Conclude that the input to the Sboxes from the original plaintext was:

Input= [0, 1, 1, 1] [1, 1, 0, 1

(Concatenate the remaining values for the input to Sbox 1 and 2.) We can now recover the value of K3 by xoring this string with the value of E(L3):

E(L3): [1, 0, 1, 0]

⊕ Input: [0, 1, 1, 1]] =K3 : [1011]

**Part 2:** Use a meet in the middle attack to recover the two keys K1 and K2 used in an implementation of

2SDES (Double encryption with SDES) using 4 rounds.

First you encrypt P=[0,1,0,1,0,1,0,1,0,1,0,1] and get C=[0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0].

**a.** Use a brute force attack to find all possible values of K1 and K2. How many seconds does it take?

CPU time: 328.28 s

**b.** Use the Meet-In-The-Middle attack to find the same information. Are they the same as the ones you found by brute force?

Yes They are the same ones

**c.** How many seconds did this take?

CPU time: 0.51 s

**d.** Explain briefly why this was so much faster. How many encryptions are required in total when doing brute force? How many encryptions/decryptions are required when performing a meet in the middle attack?

(Recall, the keys have 9 bits…) How many times faster would you expect meet in the middle to be?

In meet in the middle the hacker is able to able to obtain information from both sides. The information obtained from both sides makes it easier for the attacker. It is significantly faster than the brute force attack

Now you find that encrypting P\*=[0, 0, 1, 0, 0, 1, 0, 0,1, 0, 0, 1] with the same keys produces the ciphertext C\*=[1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0].

**e.** Repeat the meet in the middle attack, and compare the pairs of keys you got using P and C to obtain the binary values of K1 and K2:

**f.** Use the int2bin function to convert these numbers back into binary and record them here: K1= 10

K2= 13