Lab 05 - Stream Ciphers and Linear Feedback Shift Registers

Part 01

1. Encrypted message from part01\_plainText.txt by using a 4-bit LFSR with the seed “1001” with the first and last bit XORd with each other.

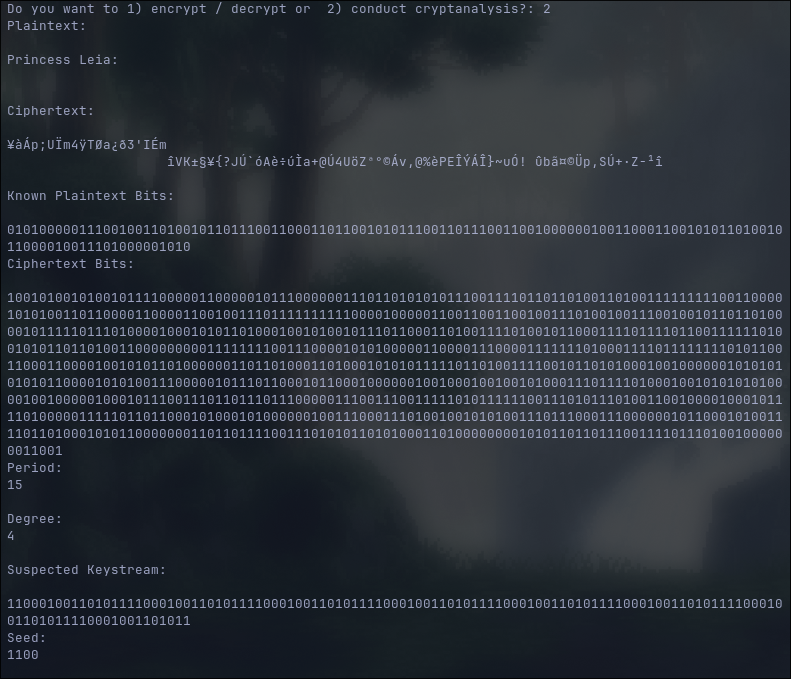
(10 points)

* 

Part 02

1. XOR the known plaintext with the first X bits of the ciphertext, where X is the number of bits in the known plaintext

(5 points)

* 

1. Calculate the period by showing the number of bits that are repeated in the keystream

(5 points)

* period = 15 (110001001101011)

1. Calculate the degree (Show your work step by step)

(5 points)

* period = 15 (110001001101011) repeating
* degree = log\_2(15 + 1) = 4

1. Calculate the seed by pasting your keystream into a spreadsheet and tracing it back to the initial starting values

(10 points)

* seed = 1100

# 0 0 (1 1) -> 0

# 0 0 0 1 -> 1

# 1 0 0 0 -> 0

# 0 1 0 0 0

# 0 0 1 0 1

# 1 0 0 1 1

# 1 1 0 0 0

# 0 1 1 0 1

# 1 0 1 1 0

# 0 1 0 1 1

# 1 0 1 0 1

# 1 1 0 1 1

# 1 1 0 1

# 1 1 0

# 1 0

# seed = 1100

1. Use the spreadsheet from step 4 to find the LFSR configuration and show your work for each step.

(10 points)

# Base

# 0 0 1 1

# 0 0 0 1

# 1 0 0 0

# 0 1 0 0

# 1 1

# 0

#

#

#

# 0 1

# 1

#

#

#

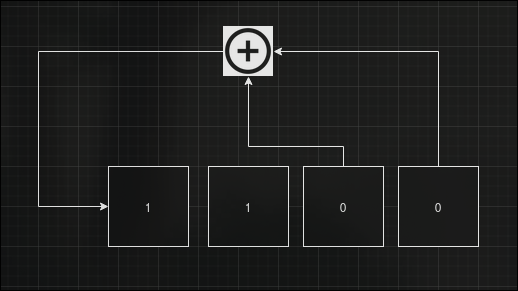
# 0 0

# 0

And so on

1. Draw a diagram of the LFSR configuration you identified in step 5

(10 points)

* 

1. Translate the plaintext into ASCII

(15 points)

* Princess Leia:

Why, you stuck up, half-witted, scruffy-looking nerf-herder.  
  
Han Solo:

Who’s scruffy-looking?

1. Upload your code to canvas as “Lab5\_part01\_02.py”. A TA will run your program and make sure it can decrypt the ciphertext. Double check that it runs without any issues before uploading it. Be sure to have both lines in the cycle() to demonstrate the part01 bits and the part02 bits.

(30 points)