Assignment 2

This assignment is to be done in pairs.

Part 1 Block Cipher

Perform the symmetric key encryption based SEED Lab:

https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_Encryption/

The detailed tasks are under the Tasks [PDF] link on the page.

Deliverables:

Complete the implementation and submit a detailed report, including screenshots, of all the steps involved. Submitting a report with only screenshots and no description will receive zero credit.

Part 2 Stream Cipher

In this problem, we will study LFSRs in somewhat more detail.

Part-A: LFSRs come in three flavors:

- LFSRs which generate a maximum-length sequence. These LFSRs are based on *primitive polynomials*.
- LFSRs which do not generate a maximum-length sequence but whose sequence length is independent of the initial value of the register. These LFSRs are based on *irreducible polynomials* that are not primitive. Note that all primitive polynomials are also irreducible.
- LFSRs which do not generate a maximum-length sequence and whose sequence length depends on the initial values of the register. These LFSRs are based on *reducible polynomials*.

We will study examples in the following. determine all sequences generated by

1. $x^4 + x + 1$
2. $x^4 + x^2 + 1$
3. $x^4 + x^3 + x^2 + x + 1$

Deliverables:

1. Key Stream Analysis:
   a. Write a generic software program in C++ or Python that accepts polynomial positions for any given polynomial (e.g., 4, 1 for X^4+X+1) along with an initial value and generates a sequence three times the maximum length, i.e., 3(2^n - 1).
   b. Generate a sequence of all the above three polynomials and analyze them according to the above-mentioned LFSR flavors.
   c. Draw the corresponding Linear Feedback Shift Register (LFSR) for each of the three polynomials.

d. Determine which of the given polynomials is primitive, which is only irreducible, and which one is reducible.
e. Repeat this whole for 3x polynomials of your choice and share analysis details as per LFSR flavors.

Part-B: Part-B: Design a synchronous stream cipher with any primitive polynomial with the highest degree (n) greater or equal to 4. The cipher takes input as ASCI, converts it into Binary, and encrypts it with the key stream.

Deliverables:

a. Synchronous Stream cipher code with comments
b. Input and output
c. Detail of Plain text attack on the cipher to find Feedback polynomial