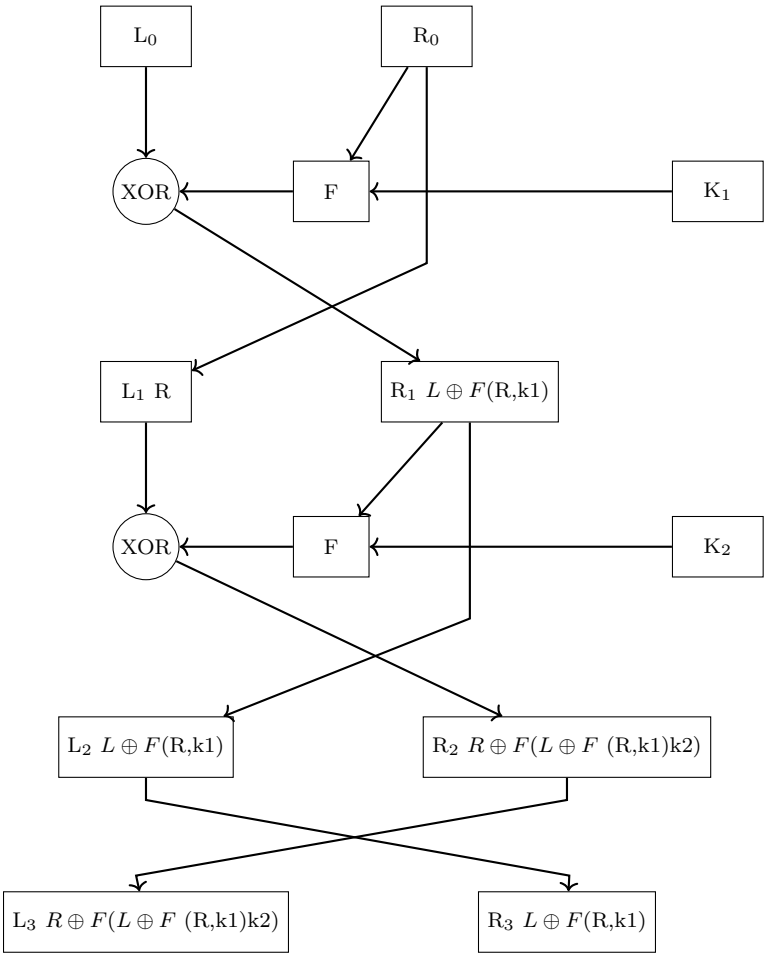
Assignment 1

Course Number: SPR200
Course Name: Basic Cryptography
Course Section: NAA
Assignment Number: 1
Student Full Name: Zakariya Outbih
Student GitHub Username: zoutbih_seneca

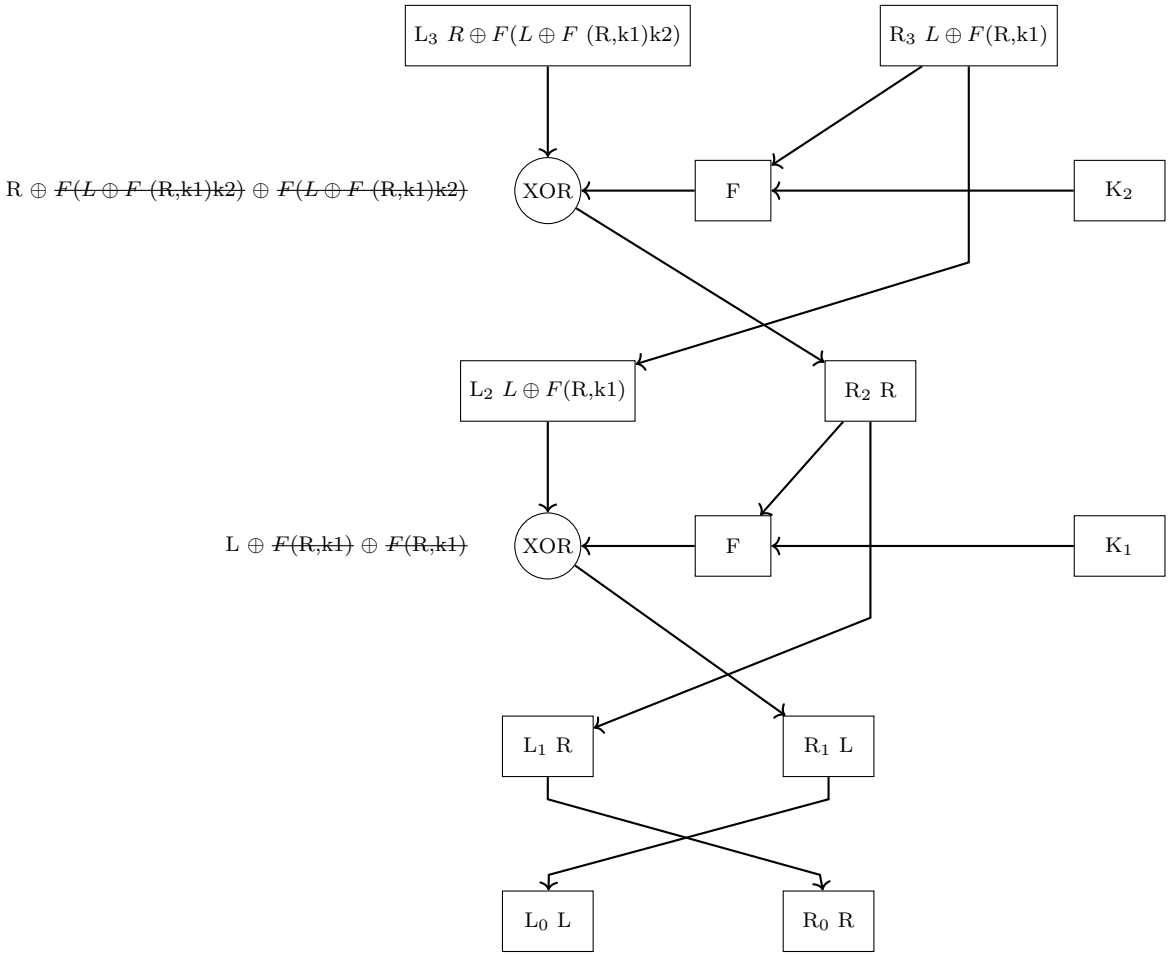
Feistel Cipher Diagrams

Feistel Cipher Encryption Diagram



I first made the encryption diagram to better understand the decryption one. Essentially there is one plaintext given and divided into two parts, L0 and R0. Each round the right side is passed through the function and the left side is xored to the output. The important thing to note is that each round R1, R2, R3 the left and right sides are swapped so R1 becomes L1 and so on.

Feistel Cipher Decryption Diagram



As you can see this diagram (decryption) is essentially the reverse of the Encryption diagram. This is how the Feistel cipher works. It uses the properties of the XOR operation to encrypt and decrypt. You can pass the plain text as many times as you would like through the function + XOR operation and still be able to decrypt the plaintext. In the XOR operation both inputs must be different for the output to be true or (1). Meaning that if the inputs are the same (or part of the inputs are the same) like in the decryption diagram then they cancel each other out.

Here is a great link from computerphile explaining the feistel cipher in more detail: Click Here;

2-DES and 3-DES: Diagrams and Explanation

2-DES:

Why 2-DES is insecure:
Although 2-DES uses two keys and seems to offer 112 bits of security ($56\ bits \times 2$), it is vulnerable to a **meet-in-the-middle attack**, which reduces the effective security to about 2^{57} operations. The attacker can encrypt the plaintext with all possible K_1 values and decrypt the ciphertext with all possible K_2 values, then match the intermediate results. This makes 2-DES only marginally more secure than DES.

3-DES (Encrypt-Decrypt-Encrypt):

Why 3-DES is better:
3-DES increases the complexity significantly by applying DES three times — Encrypt with K_1, Decrypt with K_2, then Encrypt with K_1 again. This makes it resistant to the meet-in-the-middle attack and gives an effective security of approximately 112 bits. The use of decryption in the middle also ensures backward compatibility with single DES when $K_1 = K_2$.