INF143A 22V Applied Cryptography
First mandatory assignment

Nikolay Kaleyski

February 15, 2022

1 General information

- **Deadline:** The assignment is due on March 8, 2022.
- **Submission:** A copy of the solution should be uploaded to mitt, under the "Assignments" tab.
- **Score:** The mandatory assignment accounts for 15% of the final grade.
- **Collaboration:** You can freely discuss the assignment with each other, but the solutions must be prepared individually; plagiarism will result in all involved parties failing the assignment.

2 Problems

Problem 1. ***10%*** *What is the maximum possible cycle length (that is, the number of outputs that it can produce before it loops) of an LFSR on 100 bits? Construct such an LFSR (you can give your answer either as a diagram or a polynomial).*

Problem 2. ***30%*** *Write an implementation of the Trivium stream cipher, as illustrated below. The symbol $\oplus$ denotes binary XOR, while $\odot$ denotes binary AND. The device is initialized as described in the lecture slides, that is:*

- *the 80-bit key is loaded into the first 80 bits of the 84-degree LFSR (the "first" bits here refers to the left-most 80 bits);*
- *the 80-bit initialization vector is loaded into the first 80 bits of the 93-degree LFSR;*
- *the right-most 3 bits of the 111-degree LFSR are set to 1;*
- *all remaining bits are set to 0.*

Recall that the cipher performs 1152 "warm-up" steps before beginning to produce output.

Use your implementation with key $K = (1, 0, 1, 0, \ldots, 1, 0)$ (80 alternating ones and zeros) and $IV = (0, 1, \ldots, 0, 1)$ (80 alternating zeros and ones) to generate 1000 *bits of keystream.*

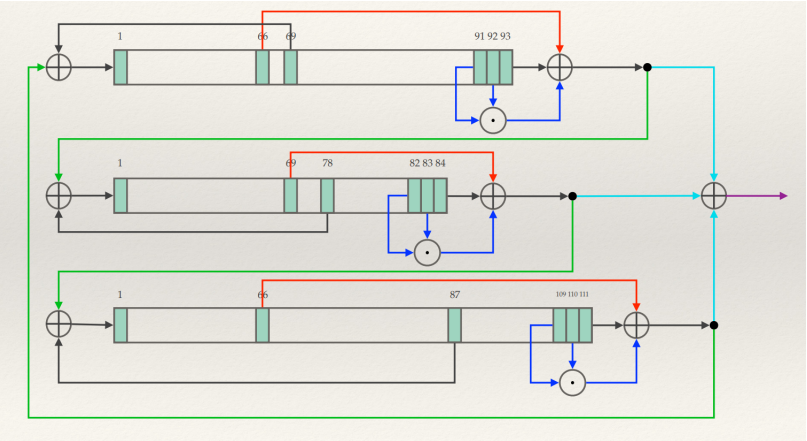

Figure 1: Trivium cipher

Problem 3. ***30%*** *Consider a simple block cipher based on a Feistel network. The size of the block is 64 bits, and the round function $F(x)$ is given by*

$$F(x,k) = x^2k + xk^2,$$

with multiplication performed in the finite field $\mathbb{F}_{2^{32}}$ given by the irreducible polynomial $p(x) = x^{32} + x^{15} + x^9 + x^7 + x^4 + x^3 + 1$, where x is the right half of the round input, and k is the round key. The cipher consists of 8 rounds. The key K is 32 bits long, and the round keys are derived from it by 4-bit cyclic shifts, i.e. the first round key is $k_1 = K$; the second round key k_2 is obtained by cyclically shifting K 4 bits to the right; and so forth.

Write an implementation of this cipher, and use the key $K = 3ACDDEF2$ (given in hexadecimal) to encrypt the plaintext $P = 1F2A0E341F2A0E34$, and to decrypt the ciphertext $C = AAAAAAAAAAAAAAAA$.

Problem 4. ***30%*** *The Python bytecode files enc.pyc and dec.pyc contain an implementation of the enctyption and decryption procedure for a simple cipher based on a Feistel network. The cipher takes 32 bits as input, and uses a 16-bit key. You should treat the exact form of the cipher as a "black box", i.e. you do not need to know how exactly the cipher functions.*

1. *Suppose that you have intercepted the plaintext*

$$P = 10101010101010101010101010101010$$

and the corresponding ciphertext

$$C = 11110100011101010101100111111110.$$

Find the 16-bit secret key by exhaustive search.

2. *Consider now a version of the algorithm in which encryption is performed twice with two different keys; i.e. the key is now 32 bits and of the form $K = (K_1, K_2)$ where both K_1 and K_2 are 16 bits long; the plaintext is first encrypted with K_1, and then the result of this encryption is encrypted*

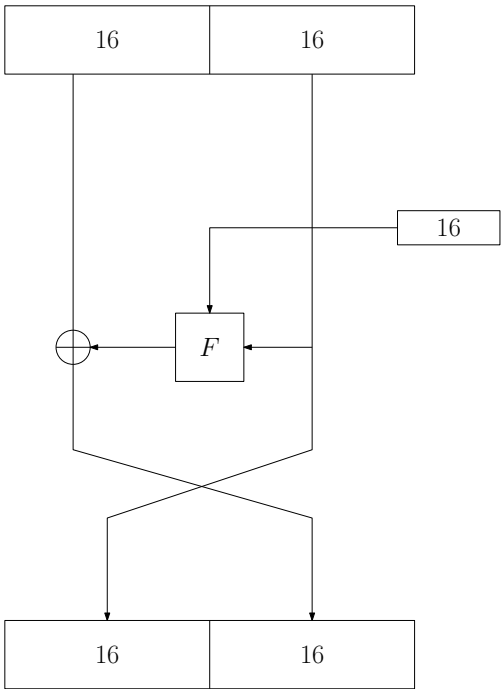

Figure 2: Feistel cipher with a block of size 32

again with K_2 to obtain the ciphertext. Suppose you have observed the same plaintext as above, i.e.

$$P = 10101010101010101010101010101010$$

and the corresponding ciphertext

$$C = 11000110100010110110000001101110.$$

Recover the 32-bit secret key $K = (K_1, K_2)$ via a meet-in-the-middle attack.