Introduction to Cryptography: Homework 2

September 22, 2021

Requirements about the delivery of this assignment:

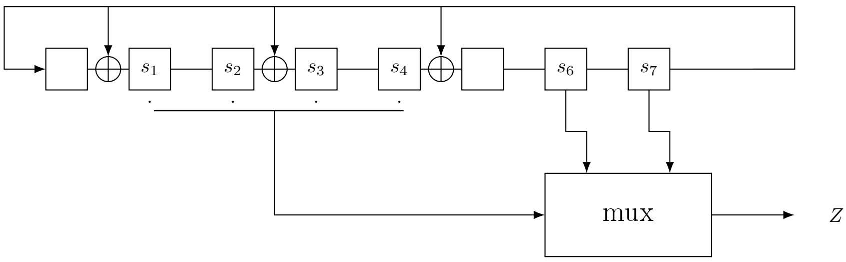
- Submit a pdf-document via Brightspace;
- Any additional files (e.g., Python scripts) can be added as well;
- Make sure that you write both name and student number on all documents (not only in the file name).

Deadline: Monday, October 4, 17:00 sharp!

Grading: You can score a total of 100 points for the hand-in assignments. To get full points, please **explain all answers clearly**.

Exercises:

1. **Filtered LFSR – Guess-and-determine attack.** In this exercise, we will break a filtered LFSR by performing a guess-and-determine attack. Before you start doing any computations, please read the entire text of the exercise (including parts (b),(c) etc.). Consider then the following filtered LFSR:



for which the multiplexer function is given by $z = s_A$ with $A = 1 + s_6 + 2s_7$. You are given the output stream $(z_0, \ldots, z_9) =$ `1111000011`.

(a) Give an upper bound on the security strength of this filtered LFSR by only considering an exhaustive key search.

(b) Mount the guess-and-determine attack to find the initial state (i.e., the key). Start by making the guesses $s_5 = s_7 = 1$ and $s_6 = 0$. (Note that this is correct.) When you arrive at a conclusion, make sure that all other leaves have terminated and the last one yields the entire given output stream. [You do not need to code for this, it is short enough to do by hand. If you nevertheless do code it, you *have* to include your own code while uploading on Brightspace and it should be clearly commented and easy to read.]

(c) Give the (full) binary tree that corresponds with the guesses that you've made in (c). [You can use the LaTeX-file as a start to typeset the search tree. An example of a search tree can be found in the lecture notes (for a bigger case). The more structured your approach, the less room for error.]

(d) What is the total number of guesses that you had to make? [Do not include the guesses $s_5 = s_7 = 1$ and $s_6 = 0$.]

(e) Assume that you would have performed the guesses for

$$(s_5, s_6, s_7) \in \{(0,0,0), (0,0,1), (0,1,0), (1,0,0), (0,1,1), (1,1,0), (1,1,1)\}$$

first. For each triple (s_5, s_6, s_7) of options, you had to make some number of guesses to get to a contradiction, thus eliminating the triple as viable option. You may assume that, for each triple, this number is the same as what you found in (d). Based on the guess-and-determine attack, give an upper bound on the security strength of this filtered LFSR.

2. **Security strength of a linear cipher.** The CryptoMatrix-64 cipher encrypts 64-bit messages, using a $4096(= 64^2)$-bit key. We interpret this 4096-bit key K as a 64×64 matrix, where the first row contains $K_0, \ldots, K_{63}$, the second row contains $K_{64}, \ldots, K_{127}$, etc. The encryption of a 64-bit message $P = (P_0, \ldots, P_{63})$ is given by

$$C = \text{CryptoMatrix}_K(P) = K \cdot P$$

and decryption of a ciphertext $C = (C_0, \ldots, C_{63})$ is then given by

$$P = \text{CryptoMatrix}_K^{-1}(C) = K^{-1} \cdot C.$$

Note that not all keys K yield an invertible matrix in this way. Therefore only keys K such that K^{-1} may count for this cipher. By a combinatorics argument, one can show that more than 25% of all keys yield an invertible matrix.

   (a) Suppose that actually exactly 25% of the keys yield an invertible matrix. What is the upper bound on the security strength based on an exhaustive key search?

   (b) Show/explain that encryption (and decryption) is a linear operation.

   (c) Show that by choosing 64 plaintexts (and doing 64 encryptions) one can retrieve the entire key.

   (d) Based on (c), give a new upper bound on the security strength of the CryptoMatrix cipher.

3. **Distinguishing a stream cipher.** In this exercise, we consider a stream cipher BYTTASource that generates as keystream a sequence of bytes. It has the particular property that all these bytes are different from `0x00`. We want to find a distinguisher $\mathcal{D}$ that has non-negligible advantage in distinguishing BYTTASource from its ideal version.

   (a) What is the ideal version of BYTTASource, or in general, a stream cipher?

   (b) Give a distinguisher $\mathcal{D}$ that distinguishes BYTTASource from its ideal version. [Make sure that it will yield a non-negligible advantage in (d).]

   (c) What is the probability that your distinguisher guesses that it is talking to BYTTASource, while it is actually talking to the ideal version, given that the distinguisher gets M bytes of keystream?

   (d) Give the advantage of your distinguisher.

   (e) Explain why this advantage gets close to 1, if M becomes larger.

   (f) What is the security strength of BYTTASource? [Minimze $M/\,\text{Adv}$ where M is the number of output bytes and Adv is the advantage from (d).]

Hand in assignment:

1. **(100 points) Combiner LFSR – Divide-and-conquer attack** In this exercise, we will explore the divide-and-conquer attack on a combiner LFSR.

   Consider the combiner LFSR in Figure 1.

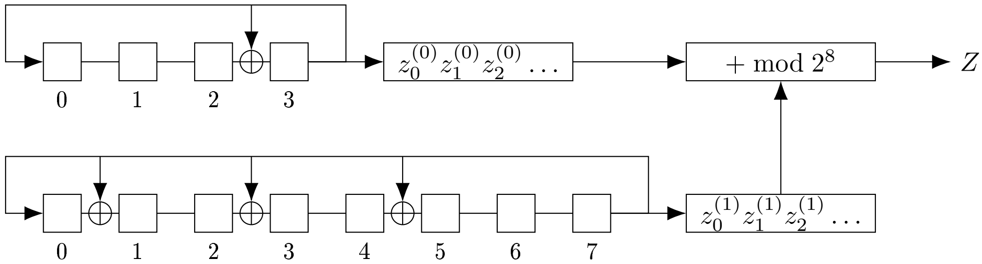


Figure 1: A combiner LFSR.

   For addition, we take the $z_0^{(0)}$ and $z_0^{(1)}$ to be the most significant bits. So, for instance, 00000001 is 1, and 01000000 is 64. The output bytes are given as numbers from 0 to 255.

   (a) Give an upper bound on the security strength of this combiner LFSR by only considering an exhaustive key search. 10 pt

   (b) Let the output stream of the combiner LFSR be given by $Z = (145, 68)$. We guess that the initial state for the 4-bit LFSR is `0101`. Compute the first sixteen bits of the output stream for this 4-bit LFSR (using this initial state). 20 pt

   (c) Assume that the guess of `0101` for the 4-bit LFSR is correct. Give the first sixteen bits of the output stream for the 8-bit LFSR. 10 pt

   (d) Mount a linear attack on the 8-bit LFSR to obtain its current state (after 8 iterations). [You only need the first 8 output bits. Since it is not decimated, you can use the method from Assignment 1 Exercise 2 (Linear Feedback Shift Register – basics.).] 20 pt

   (e) Explain why your guess `0101` of the initial state was not correct. 10 pt

   (f) Perform the full divide-and-conquer attack on this combiner LFSR. [There is no need to do this by hand, but please provide clear references and/or code.] 30 pt