INF143A 23V Applied Cryptography
First mandatory assignment

Nikolay Kaleyski

1 General information

- **Deadline:** The assignment is due on **March 9, 2023**.
- **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.
- **Passing grade:** You must obtain at least 20% to successfully pass the assignment.

2 Problems

Problem 1. ***20%*** *Construct an LFSR that will go through* $2^{30}-1$ *distinct states before it loops. You can specify the LFSR as either a diagram or a recurrence relation.*

Output the first 500 output bits of the LFSR if it starts in the initial state

$$(1, 0, 0, \ldots, 0, 0, 0, 1).$$

Please provide your output in the format of a text file of zeros and ones, with no spaces, commas, or other delimiters in between.

Sample output: *See* `lfsr_sample.txt` *for the first 10 bits of a degree* 4 *LFSR producing a cycle of* 15 *states and starting from the state* $(1, 0, 0, 1)$.

Problem 2. ***30%*** *Consider the finite field* $\mathbb{F}_{2^6}$ *generated by the primitive polynomial* $p(x) = x^6 + x + 1$. *Suppose that we have a function*

$$F(x, k) = x^3 + (x + k)^3 + k,$$

where the operations used are finite field addition, multiplication, and exponentiation. The inputs x *and* k *are 6-bit blocks, and the output of the function is another 6-bit block.*

Write a program that will compute the value of the function for any inputs x *and* k, *e.g. which for* $x = (0, 0, 0, 0, 0, 0)$ *and* $k = (0, 0, 0, 0, 0, 0)$ *will output* $(0, 0, 0, 0, 0, 0)$.

Use this program to generate a lookup table of the function, i.e. a file containing a list of pairs of the form $x, k- > f(x)$ for all possible 6-bit inputs x and k.

Sample output: *See* `ff_sample.out` *for a file containing the truth table of the same function $F(x,k) = x^3 + (x+k)^3 + k$ but for the field $\mathbb{F}_{2^3}$ (and therefore for 3-bit input and output blocks).*

Problem 3. ***50 %*** *Consider the block cipher inspired by the lightweight Simon cipher [1] depicted in Figure 1:*

- *the block size is 16 bits, e.g. L_i and R_i are 8 bits long;*
- *S^k denotes a left cyclic shift by k bits;*
- *$\oplus$ denotes XOR;*
- *"AND" denotes logical "and";*
- *K_i is the round key, which is 8 bits long.*

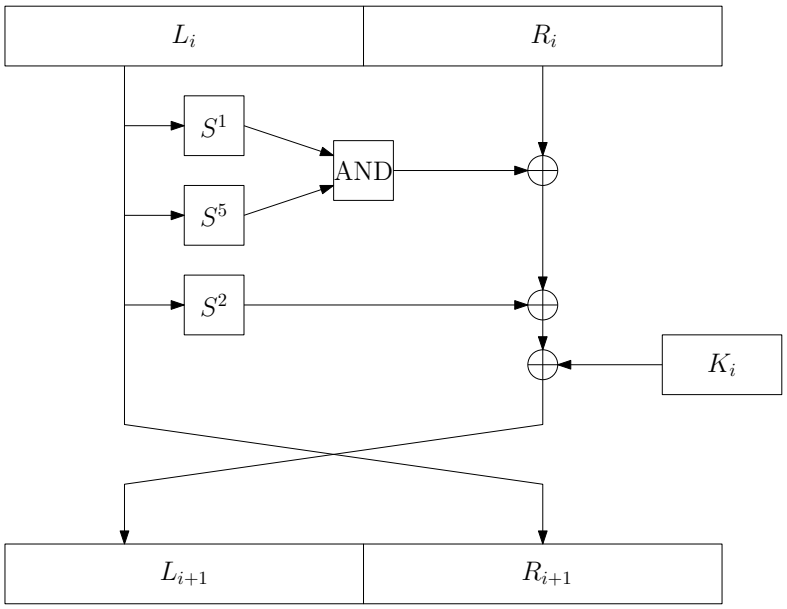

Figure 1: A 4-round toy cipher based on a Feistel network

The cipher consists of 4 rounds. The key K is 32 bits long, and the round keys K_0, K_1, K_2, K_3 are such that $K = K_0||K_1||K_2||K_3$, where $||$ denotes concatenation[1]*.*

Write an implementation allowing one to encrypt and decrypt with this toy cipher. Your implementation should provide a function that takes as input a 16-bit block of plaintext and a 32-bit key, and outputs a 16-bit block of ciphertext.

[1]Note that this is a simplified construction for INF143A, and key schedules like this where the keys are disjoint, are not secure!

The blocks and keys can internally be represented in any way that you find suitable, but your implementation should read the input from a text file containing 16 digits (0 or 1), and should record the output in the same format.

Sample output: *The files* `cipher_in1.txt` *and* `cipher_in2.txt` *contain two sample input (plaintext) blocks, while* `cipher_out1.txt` *and* `cipher_out2.txt` *contain the corresponding output (ciphertext) blocks. Both pairs of plaintext-ciphertext are encrypted using the key in* `cipher_key.txt`*.*

NB: *When implementing Feistel networks, be careful about the order of the left and right blocks at the output. The blocks are typically swapped one extra time after the last round in order to make decryption work correctly. If you are having trouble getting the correct input for the provided samples, try swapping the left and right blocks before or after the entire Feistel network.*

References

[1] Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L. The SIMON and SPECK lightweight block ciphers. In Proceedings of the 52nd annual design automation conference 2015 Jun 7 (pp. 1-6).