TMA4155 — Exercise 3

Kristian Gjøsteen

2006-09-14

Task 1 — RC4-like stream cipher

We can make an RC4-like stream cipher by replacing 255 and 256 with 15 and 16 in RC4, as follows:
Input: $K \in \{0 \ldots 15\}^{15}$, n.

1. For $i = 0$ to 15, do: $S_i \leftarrow i$.
2. $j \leftarrow 0$.
3. For $i = 0$ to 15, do:
   (a) $j \leftarrow (j + S_i + K_i) \bmod 16$.
   (b) Swap S_i and S_j.
4. $i \leftarrow 0$, $j \leftarrow 0$.
5. For $i = 0$ to $n-1$, do:
   (a) $i \leftarrow (i+1) \bmod 16$.
   (b) $j \leftarrow (j + S_i) \bmod 16$.
   (c) Swap S_i and S_j.
   (d) $t \leftarrow (S_i + S_j) \bmod 16$.
   (e) $r_i \leftarrow S_t$.

Output: $r = (r_0, r_1, \ldots, r_{n-1}) \in \{0, \ldots, 15\}^n$.

a. Find the first 5 digits of the output when $K = (1, 1, 1, \ldots, 1)$.

Task 2 — Counter mode

Use the toy block cipher from Exercise 2, Task 1b in counter mode to generate 15 bits. Let the initialization vector be 0000 0000 and the key be 1011 1101.

Task 3 — LFSR

Consider the linear feedback shift register of length 8 with linear feedback relation $x_n = x_{n-2} \oplus x_{n-3} \oplus x_{n-7} \oplus x_{n-8}$, and the filter function $f(x_n, \ldots, x_{n-7}) = x_{n-1} \wedge x_{n-3} \wedge x_{n-5} \wedge x_{n-6} \oplus x_{n-1} \oplus 1$.

We create a simple stream cipher as follows: The key is 8 bits long, as is the initialization vector. The initial content of the LFSR is the xor of the key and the iv. If the number of ones in the shift register is even, the x_0 bit is xored with 1. To generate a bit, we first step the LFSR, then use the filter function to compute the bit.

a. Generate eight bits of output using key 0000 1111 and initialization vector 1111 0101.

a. Show how you can with high probability extract the initial state of the shift register using eight bits of output. (Hint: Compare the values of the filter function to $g(x_n, \ldots, x_{n-7}) = x_{n-1} \oplus 1$.)

Task 4 — Modular arithmetic

a. Compute $x^6 \bmod 7$ for $x = 1, 2, \ldots, 6$.

b. Compute $x^8 \bmod 9$ for $x = 1, 2, \ldots, 8$.