Problems

2.1. The stream cipher described in Definition 2.1.1 can easily be generalized to work in alphabets other than the binary one. For manual encryption, an especially useful one is a stream cipher that operates on letters.

1. Develop a scheme which operates with the letters A, B,…, Z, represented by the numbers 0,1,…,25. What does the key (stream) look like? What are the encryption and decryption functions?
2. Decrypt the following cipher text:
   `bsaspp kkuosp`
   which was encrypted using the key:
   `rsidpy dkawoa`
3. How was the young man murdered?

2.2. Assume we store a one-time key on a CD-ROM with a capacity of 1 Gbyte. Discuss the *real-life* implications of a One-Time-Pad (OTP) system. Address issues such as life cycle of the key, storage of the key during the life cycle/after the life cycle, key distribution, generation of the key, etc.

2.3. Assume an OTP-like encryption with a short key of 128 bit. This key is then being used periodically to encrypt large volumes of data. Describe how an attack works that breaks this scheme.

2.4. At first glance it seems as though an exhaustive key search is possible against an OTP system. Given is a short message, let's say 5 ASCII characters represented by 40 bit, which was encrypted using a 40-bit OTP. Explain *exactly* why an exhaustive key search will not succeed even though sufficient computational resources are available. This is a paradox since we know that the OTP is unconditionally secure. That is, explain why a brute-force attack does not work.

Note: You have to resolve the paradox! That means answers such as "The OTP is unconditionally secure and therefore a brute-force attack does not work" are not valid.

2.5. We will now analyze a pseudorandom number sequence generated by a LFSR characterized by $(c_2 = 1, c_1 = 0, c_0 = 1)$.

1. What is the sequence generated from the initialization vector $(s_2 = 1, s_1 = 0, s_0 = 0)$?
2. What is the sequence generated from the initialization vector $(s_2 = 0, s_1 = 1, s_0 = 1)$?
3. How are the two sequences related?

2.6. Assume we have a stream cipher whose period is quite short. We happen to know that the period is 150–200 bit in length. We assume that we do *not* know anything else about the internals of the stream cipher. In particular, we should not assume that it is a simple LFSR. For simplicity, assume that English text in ASCII format is being encrypted.

Describe in detail how such a cipher can be attacked. Specify exactly what Oscar has to know in terms of plaintext/ciphertext, and how he can decrypt all ciphertext.

2.7. Compute the first two output bytes of the LFSR of degree 8 and the feedback polynomial from Table 2.3 where the initialization vector has the value FF in hexadecimal notation.

2.8. In this problem we will study LFSRs in somewhat more detail. LFSRs come in three flavors:

- LFSRs which generate a maximum-length sequence. These LFSRs are based on *primitive polynomials*.
- LFSRs which do not generate a maximum-length sequence but whose sequence length is independent of the initial value of the register. These LFSRs are based on *irreducible polynomials* that are not primitive. Note that all primitive polynomials are also irreducible.
- LFSRs which do not generate a maximum-length sequence and whose sequence length depends on the initial values of the register. These LFSRs are based on *reducible polynomials*.

We will study examples in the following. Determine *all* sequences generated by

1. x^4+x+1
2. x^4+x^2+1
3. $x^4+x^3+x^2+x+1$

Draw the corresponding LFSR for each of the three polynomials. Which of the polynomials is primitive, which is only irreducible, and which one is reducible? Note that the lengths of all sequences generated by each of the LFSRs should add up to 2^m-1.

2.9. Given is a stream cipher which uses a single LFSR as key stream generator. The LFSR has a degree of 256.

1. How many plaintext/ciphertext bit pairs are needed to launch a successful attack?
2. Describe all steps of the attack in detail and develop the formulae that need to be solved.
3. What is the key in this system? Why doesn't it make sense to use the initial contents of the LFSR as the key or as part of the key?

2.10. We conduct a known-plaintext attack on an LFSR-based stream cipher. We know that the plaintext sent was:

```
1001 0010 0110 1101 1001 0010 0110
```

By tapping the channel we observe the following stream:

```
1011 1100 0011 0001 0010 1011 0001
```

1. What is the degree m of the key stream generator?
2. What is the initialization vector?
3. Determine the feedback coefficients of the LFSR.

4. Draw a circuit diagram and verify the output sequence of the LFSR.

2.11. We want to perform an attack on another LFSR-based stream cipher. In order to process letters, each of the 26 uppercase letters and the numbers 0, 1, 2, 3, 4, 5 are represented by a 5-bit vector according to the following mapping:

$$\begin{aligned} A &\leftrightarrow 0 = 00000_2 \\ &\vdots \\ Z &\leftrightarrow 25 = 11001_2 \\ 0 &\leftrightarrow 26 = 11010_2 \\ &\vdots \\ 5 &\leftrightarrow 31 = 11111_2 \end{aligned}$$

We happen to know the following facts about the system:

- The degree of the LFSR is $m = 6$.
- Every message starts with the header `WPI`.

We observe now on the channel the following message (the fourth letter is a zero):
`j5a0edj2b`

1. What is the initialization vector?
2. What are the feedback coefficients of the LFSR?
3. Write a program in your favorite programming language which generates the whole sequence, and find the whole plaintext.
4. Where does the thing after `WPI` live?
5. What type of attack did we perform?

2.12. Assume the *IV* and the key of Trivium each consist of 80 all-zero bits. Compute the first 70 bits $s_1, \ldots, s_{70}$ during the warm-up phase of Trivium. Note that these are only internal bits which are not used for encryption since the warm-up phase lasts for 1152 clock cycles.