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?

2.1

1. $y_i = x_i + K_i \bmod 26$
   $x_i = y_i - K_i \bmod 26$
   The keystream is a sequence of random integers from Z_{26}.
2. $x_1 = y_1 - K_1 = \text{"}B\text{"} - \text{"}R\text{"} = 1 - 17 = -16 \equiv 10 \bmod 26 = \text{"}K\text{"}$ etc $\cdots$
   Decrypted Text: "KASPAR HAUSER"
3. He was knifed.

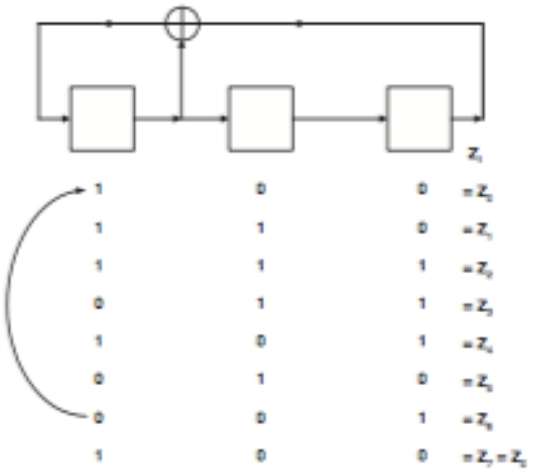
2.3. Assume an OTP-like encryption with a short key of 128 bit. This key is then being used periodically to encrypt large volumes of data. Describe how an attack works that breaks this scheme.

2.3
We need 128 pairs of plaintext and ciphertext *bits* (i.e., 16 byte) in order to determine the key. s_i is being computed by
$s_i = x_i \oplus y_i;\ \ i = 1, 2, \cdots, 128.$

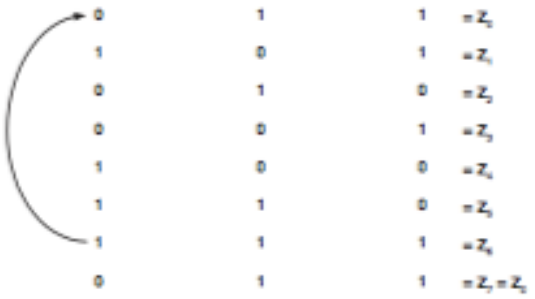
2.5. We will now analyze a pseudorandom number sequence generated by a LFSR characterized by $(c_2 = 1, c_1 = 0, c_0 = 1)$.

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?

2.5



1. Sequence 1: z_i = 0 0 1 1 1 0 1 0 0 1 1 1 0 1 ...



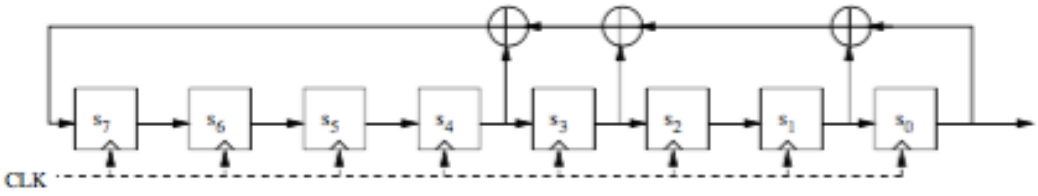
2. Sequence 2: z_i = 1 1 0 1 0 0 1 1 1 0 1 0 0 1 ...

3. The two sequences are shifted versions of one another.

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.

2.7

The feedback polynomial from 2.3 is $x^8+x^4+x^3+x+1$.



1	1	1	1	1	1	1	1	$=Z_0$
0	1	1	1	1	1	1	1	$=Z_1$
0	0	1	1	1	1	1	1	$=Z_2$
0	0	0	1	1	1	1	1	
0	0	0	0	1	1	1	1	
1	0	0	0	0	1	1	1	
0	1	0	0	0	0	1	1	
0	0	1	0	0	0	0	1	
1	0	0	1	0	0	0	0	
1	1	0	0	1	0	0	0	
1	1	1	0	0	1	0	0	
0	1	1	1	0	0	1	0	
0	0	1	1	1	0	0	1	
1	0	0	1	1	1	0	0	
0	1	0	0	1	1	1	0	$=Z_{14}$
0	0	1	0	0	1	1	1	$=Z_{15}$

So, the resulting first two output bytes are $(1001000011111111)_2 = (90FF)_{16}$.