Solution of Problem 1

a) The bit error occurs in block $C_i$, $i > 0$, with block size BS.

<table>
<thead>
<tr>
<th>mode</th>
<th>$M_i$</th>
<th>max #err</th>
<th>remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECB</td>
<td>$E_K^{-1}(C_i)$</td>
<td>BS</td>
<td>only block $C_i$ is affected</td>
</tr>
<tr>
<td>CBC</td>
<td>$E_K^{-1}(C_i) \oplus C_{i-1}$</td>
<td>BS+1</td>
<td>$C_i$ and one bit in $C_{i+1}$</td>
</tr>
<tr>
<td>OFB</td>
<td>$C_i \oplus Z_i$</td>
<td>1</td>
<td>one bit in $C_i$, as $Z_0 = C_0, Z_i = E_K(Z_{i-1})$</td>
</tr>
<tr>
<td>CFB</td>
<td>$C_i \oplus E_k(C_{i-1})$</td>
<td>BS+1</td>
<td>$C_i$ and one bit in $C_{i+1}$</td>
</tr>
<tr>
<td>CTR</td>
<td>$C_i \oplus E_K(Z_i)$</td>
<td>1</td>
<td>one bit in $C_i$, $Z_0 = C_0, Z_i = Z_{i-1} + 1$</td>
</tr>
</tbody>
</table>

b) If one bit of the ciphertext is lost or an additional one is inserted in block $C_i$ at position $j$, all bits beginning with the following positions may be corrupt:

<table>
<thead>
<tr>
<th>mode</th>
<th>block</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECB</td>
<td>$i$</td>
<td>1</td>
</tr>
<tr>
<td>CBC</td>
<td>$i$</td>
<td>1</td>
</tr>
<tr>
<td>OFB</td>
<td>$i$</td>
<td>$j$</td>
</tr>
<tr>
<td>CFB</td>
<td>$i$</td>
<td>$j$</td>
</tr>
<tr>
<td>CTR</td>
<td>$i$</td>
<td>$j$</td>
</tr>
</tbody>
</table>

In ECB and CBC, all bits of blocks $C_i, C_{i+1}$ may be corrupt.
In OFB, CFB, CTR, all bits beginning at position $j$ of block $C_i$ may be corrupt.
Solution of Problem 2

The given AES-128 key is denoted in hexadecimal representation:

\[ K = (2D\ 61\ 72\ 69\ |\ 65\ 00\ 76\ 61\ |\ 6E\ 00\ 43\ 6C\ |\ 65\ 65\ 66\ 66) \]

(a) The round key is \( K_0 = K = (W_0\ W_1\ W_2\ W_3) \) with \( W_0 = (2D\ 61\ 72\ 69) \), \( W_1 = (65\ 00\ 76\ 61) \), \( W_2 = (6E\ 00\ 43\ 6C) \), \( W_3 = (65\ 65\ 66\ 66) \).

(b) To calculate the first 4 bytes of round key \( K_1 \), recall that \( K_1 = (W_4\ W_5\ W_6\ W_7) \).

Follow Alg. 1 as given in the lecture notes to calculate \( W_4 \):

\[ \text{Algorithm 1 AES key expansion (applied)} \]

\[
\text{for } i \leftarrow 4; \ i < 4 \cdot (r + 1); \ i ++ \text{ do}
\]

\[
\text{Initialize for-loop with } i \leftarrow 4. \ \text{We have } r = 1 \text{ for } K_1.
\]

\[
\text{tmp} \leftarrow W_{i-1}
\]

\[
\text{tmp} \leftarrow W_3 = (65\ 65\ 66\ 66)
\]

\[
\text{if } (i \mod 4 = 0) \text{ then}
\]

\[
\text{result is true as } i = 4.
\]

\[
\text{tmp} \leftarrow \text{SubBytes(RotByte(tmp))} \oplus \text{Rcon}(i/4)
\]

\[
\text{Evaluate this operation step by step:}
\]

\[
\text{RotByte(tmp)} = (65\ 66\ 66\ 65), \text{ i.e., a cyclic left shift of one byte}
\]

\[
\text{To compute SubBytes(65\ 66\ 66\ 65) evaluate Table 5.8 for each byte:}
\]

\[
\text{(row 6, col 5) provides } 77_{10} = 4D_{16}
\]

\[
\text{(row 6, col 6) provides } 51_{10} = 33_{16}
\]

\[
\text{Note that the indexation of rows and columns starts with zero.}
\]

\[
\text{SubBytes(65\ 66\ 66\ 65) = (4D\ 33\ 33\ 4D)}
\]

\[
i/4 = 1
\]

\[
\text{Rcon}(1) = (RC(1) 00\ 00\ 00), \text{ with } RC(1) = x^{1-1} = x^0 = 1 \in \mathbb{F}_{2^8}.
\]

\[
\text{tmp} \leftarrow (4D\ 33\ 33\ 4D) \oplus (01\ 00\ 00\ 00) = (4C\ 33\ 33\ 4D)
\]

\[
\text{end if}
\]

\[
W_4 \leftarrow W_{i-4} \oplus \text{tmp} \ W_4 \leftarrow W_0 \oplus \text{tmp}. \text{ Then, next iteration, } i \leftarrow 5...
\]

\[
\text{end for}
\]
Solution of Problem 3

Message \( m = (m_1m_2, ..., m_l) \), with \( m_i \in \mathbb{F}_2 \).

Key \( k = (k_1k_2, ..., k_n) \), with \( k_i \in \mathbb{F}_2 \) and \( n < l \) \( \Rightarrow \) Keystream \( z = (z_1, z_2, ..., z_l) \)

\[
\begin{align*}
z_i &= k_i, \quad 1 \leq i \leq n \\
z_i &= \sum_{j=1}^{n} s_j z_{ij} \mod 2, \quad n < i \leq l \\
c_i &= z_i \oplus m_i, \quad 1 \leq i \leq l
\end{align*}
\]

a) Decryption: \( m_i = c_i \oplus z_i \)

b) If \( k = 0 = (00...0) \), it follows \( z_i = 0 \), \( 1 \leq i \leq n \), and \( z_i = 0 \), \( n < i \leq l \) and \( c_i = m_i \), \( 1 \leq i \leq l \). In this case, the plaintext is not encrypted at all.

c) key length \( n = 4 \), key \( k = (0110) \),
addition paths \( s_1 = s_4 = 1, \ s_2 = s_3 = 0 \) \( \Rightarrow s = (1001) \),
stream length \( l = 20 \)

\[
\begin{array}{cccccccccc}
z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 & z_9 & z_{10} \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & z_{17} & z_{18} & z_{19} & z_{20} \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0
\end{array}
\]

The summation simplifies to \( z_i = \sum_{j=1}^{n} s_j z_{ij} = z_{i-1} \oplus z_{i-4}, \ 4 < i \leq 20 \)
- \( n \) provide registers \( 2^n \) states
- Maximal period: \( p_{\text{max}} = 2^n - 1 = 15 \) (Minor remark: fulfilled if \( z_i \) is a primitive polynomial)
- The keystream repeats itself at \( z_{16} \)

encryption:

<table>
<thead>
<tr>
<th>( m )</th>
<th>1011</th>
<th>0001</th>
<th>0100</th>
<th>1101</th>
<th>0100</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z )</td>
<td>0110</td>
<td>0100</td>
<td>0111</td>
<td>1010</td>
<td>1100</td>
</tr>
<tr>
<td>( m \oplus z )</td>
<td>1101</td>
<td>0101</td>
<td>0011</td>
<td>0111</td>
<td>1000</td>
</tr>
</tbody>
</table>