Solution of Problem 1

Let $p = 31$, $q = 43$. As described in the script, the initial value $x_0$ of the Blum-Blum-Shub generator is computed from $x_{t+1}$.

$$d_1 = \left( \frac{p + 1}{4} \right)^{t+1} = 8^{10} \equiv 4 \pmod{(p - 1)}$$

$$d_2 = \left( \frac{q + 1}{4} \right)^{t+1} = 11^{10} \equiv 25 \pmod{(q - 1)}$$

$$u = x_{t+1}^{d_1} \equiv 1306^4 \equiv 4^4 \equiv 8 \pmod{p}$$

$$v = x_{t+1}^{d_2} \equiv 1306^{25} \equiv 16^{25} \equiv 4 \pmod{q}$$

SQM: $a = 16; k = 25 = (11001)_2; n = 43$; calculate $a^k \mod n$.

\[
\begin{array}{c|c|c|c}
\text{bit} & x & x^2 \mod 43 & ax^2 \mod 43 \\
\hline
1 & a = 16 & 41 & 11 \\
0 & 11 & 35 & - \\
0 & 35 & 21 & - \\
1 & 21 & 11 & 4 \\
\end{array}
\]

Compute the inverse $ap + bq = 1 = \gcd(p, q)$ using the Extended Euclidean Algorithm (EEA).

\[
\begin{array}{c|c|c|c|c|c|c}
\text{n} & a_n & b_n & f_n & r_n & c_n & d_n \\
\hline
0 & & & & & & \\
1 & & & & & & \\
2 & p = 43 & q = 31 & 1 & 12 & 1 & -1 \\
3 & 31 & 12 & 2 & 7 & -2 & 3 \\
4 & 12 & 7 & 1 & 5 & 3 & -4 \\
5 & 7 & 5 & 1 & 2 & -5 & 7 \\
6 & 5 & 2 & 2 & 1 & 13 & -18 \\
\end{array}
\]

With for $n \in \mathbb{N}_0$: $r_n = c_n \cdot p + d_n \cdot q$ and for $n \geq 2$:

$$a_n = f_n \cdot b_n + r_n \quad , \text{with } f_n \in \mathbb{N}, \ 0 \leq r_n < b_n$$

$$c_n = c_{n-2} - f_n \cdot c_{n-1}$$

$$d_n = d_{n-2} - f_n \cdot d_{n-1}$$

$$a_{n+1} = b_n$$

$$b_{n+1} = r_n$$
Hence, \(1 = \gcd(43, 31) = 13 \cdot 43 - 18 \cdot 31 = b \cdot q + a \cdot p\). We can calculate \(x_0\) as:

\[
x_0 = (vap + ubq) \mod n
\]

\[
\equiv 4 \cdot (-18) \cdot 31 + 8 \cdot 13 \cdot 43
\]

\[
\equiv -2232 + 4472
\]

\[
\equiv 434 + 473 \equiv 907 \pmod{1333}
\]

Compute \(x_1, \ldots, x_9\) with \(x_{i+1} = x_i^2 \mod n\).

Use the last five digits of the binary representation of \(x_i\) for \(b_i\). E.g., \(x_1 = 188_{10} = 1011100_2 \Rightarrow b_1 = 11100\). With \(m_i = c_i \oplus b_i, 1 \leq i \leq 9\), we can decipher the cryptogram.

<table>
<thead>
<tr>
<th>(i)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_i)</td>
<td>188</td>
<td>686</td>
<td>47</td>
<td>876</td>
<td>901</td>
<td>4</td>
<td>16</td>
<td>256</td>
<td>219</td>
</tr>
<tr>
<td>(c_i)</td>
<td>10101</td>
<td>01110</td>
<td>00011</td>
<td>01000</td>
<td>10111</td>
<td>00101</td>
<td>11110</td>
<td>01101</td>
<td>11000</td>
</tr>
<tr>
<td>(b_i)</td>
<td>11100</td>
<td>01110</td>
<td>01111</td>
<td>01100</td>
<td>00101</td>
<td>00100</td>
<td>10000</td>
<td>00000</td>
<td>11011</td>
</tr>
<tr>
<td>(m_i)</td>
<td>01001</td>
<td>00000</td>
<td>01100</td>
<td>00100</td>
<td>10010</td>
<td>00001</td>
<td>01110</td>
<td>01101</td>
<td>00011</td>
</tr>
</tbody>
</table>

\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline
J & A & M & E & S & B & O & N & D \\
\hline
\end{tabular}

**Solution of Problem 2**

Recall the RSA cryptosystem: \(n = pq, p \neq q\) prime and \(e \in \mathbb{Z}_{\varphi(n)}\) with \(\gcd(e, \varphi(n)) = 1\). The public key is \((n, e)\).

Our pseudo-random generator based on RSA is:

a) Select a random seed \(x_0 \in \{2, \ldots, n-1\}\).

b) Iterate: \(x_{i+1} \equiv x_i^e \mod n, i = 0, \ldots, t\).

c) Let \(b_i\) denote the last \(h\) bits of \(x_i\), where \(h = \lfloor \log_2(\log_2(n)) \rfloor\).

d) Return the pseudo-random sequence \(b_1, \ldots, b_t\) of \(h \cdot t\) pseudo-random bits.

**Solution of Problem 3**

a) With a block cipher \(E_K(x)\) with block length \(k\), the message is split into blocks \(m_i\) of length \(k\) each, \(m = (m_0, \ldots, m_{n-1})\). Take \(m = (m_0)\) and \(\hat{m} = (m_0, m_1, m_1)\) with \(m_0, m_1\) arbitrary. Then,

\[
h(\hat{m}) = E_{m_0}(m_0) \oplus E_{m_0}(m_1) \oplus E_{m_0}(m_1) = E_{m_0}(m_0) = h(m)
\]

Thus, \(h\) is neither second preimage resistant nor collision free.

Given \(y \in \mathcal{Y}\), choose \(m_0\). Then calculate

\[
c = E_{m_0}(m_0),
\]

\[
m_1 = D_{m_0}(c \oplus y).
\]
It follows that
\[
h(m_0, m_1) = E_{m_0}(m_0) \oplus E_{m_0}(D_{m_0}(c \oplus y)) = c \oplus c \oplus y = y.
\]
Hence, \( h \) is not preimage resistant, either.

b) \( \hat{h} \) replaces XOR (\( \oplus \)) by AND (\( \odot \)) and remains the same as \( h \) otherwise. Take \( m = (m_0, m_0) \), with \( m_0 \) chosen arbitrarily. Then,
\[
\hat{h} = E_{m_0}(m_0) \odot E_{m_0}(m_0) = E_{m_0}(m_0) = \hat{h}((m_0)).
\]
\( \hat{h} \) is neither second preimage resistant nor collision free.

c) The more blocks are hashed the more bits are 0.