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 $x_{t+1}$.

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

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

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

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

Compute the inverse $ap + bq = 1$ using the Extended Euclidean algorithm.

\[
egin{align*}
43 &= 31 \cdot 1 + 12 \\
31 &= 12 \cdot 2 + 7 \\
12 &= 7 \cdot 1 + 5 \\
7 &= 5 \cdot 1 + 2 \\
5 &= 2 \cdot 2 + 1 \\
1 &= 5 - 2 \cdot 2
\end{align*}
\]

\[
= 5 - 2 \cdot (7 - 5) = 3 \cdot 5 - 2 \cdot 7
\]

\[
= 3 \cdot (12 - 7) - 2 \cdot 7 = 3 \cdot 12 - 5 \cdot 7
\]

\[
= 3 \cdot 12 - 5 \cdot (31 - 12 \cdot 2) = 13 \cdot 12 - 5 \cdot 31
\]

\[
= 13 \cdot (43 - 31 \cdot 1) - 5 \cdot 31
\]

\[
= 13 \cdot 43 - 18 \cdot 31
\]

We can calculate $x_0$ as:

$$x_0 = (vap + ubq) \mod n$$

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

$\equiv -2232 + 4472$

$\equiv 2240 \equiv 907 \pmod{1333}$

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.
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:

1. Select a random seed \( x_0 \in \{2, \ldots, n - 1\} \).
2. Iterate: \( x_{i+1} \equiv x_i^e \mod n \), \( i = 0, \ldots, t \).
3. Let \( b_i \) denote the last \( h \) bits of \( x_i \), where \( h = \lfloor \log_2(\lfloor \log_2(n) \rfloor) \rfloor \).
4. 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,

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

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 \( m = (m_1, m_1) \), with \( m_1 \) chosen arbitrarily. Then,

\[
\hat{h} = E_{m_1}(m_1) \odot E_{m_1}(m_1) = E_{m_1}(m_1) = \hat{h}((m_1)).
\]

\( \hat{h} \) is neither second preimage resistant nor collision free.