1. Given a public exponent, find suitable prime factors of the public modulus.

2. Consider a modified RSA signature scheme, in which we do not rely on CRT to combine rings $\mathbb{Z}_p$ and $\mathbb{Z}_q$ into ring $\mathbb{Z}_{pq}$, but instead, we work in one ring $\mathbb{Z}_n$, where n is a sufficiently large prime. The modified scheme works as follows.

   (a) Alice selects a sufficiently large prime n, this is her public key.

   (b) Alice calculates her private exponent $d \in \mathbb{Z}_{\varphi(n)}$ such that $d \cdot e \equiv 1 \pmod{\varphi(n)}$.

   (c) To sign a document $\hat{d}$, Alice takes a hash of it $m = H(\hat{d}) \in \mathbb{Z}_n$, where H is some cryptographic hash function. Then she distributes her signature $m^d \bmod n$ along with the document $\hat{d}$.

   (d) To verify the signature, Bob computes $\left(m^d \bmod n\right)^e \bmod n = m$. The signature is valid if $m = H(\hat{d})$.

   This signature scheme is not secure against passive adversary Carol, who can create an arbitrary amount of fake signatures on behalf of Alice. How can Carol do that?

3. Let Alice sends a cryptogram $m^e \mod n$ to Bob. Can adversary Carol recover m if $m^e < n$?

4. Alice sends the same message encrypted using the RSA algorithm to three different people with public keys $n = 87, n = 115, n = 187$. Let the public exponent be 3. Adversary Carol intercepts 3 cryptograms $c_1 = 43, c_2 = 80, c_3 = 65$. Can Eve recover the message without factoring public keys?

5. Adversary Carol intercepted two RSA cryptograms, $y_1 = 537$ sent by Alice to Bob, and $y_2 = 285$ sent by Alice to Eve. Alice knows that Bob's public exponent $e_1 = 18$, and public modulus $n_1 = 943$, while Eve's public exponent $e_2 = 19$, and her public modulus $n_2 = 943$. What is the message m sent by Alice to Bob and Eve?

6. Suppose that adversary Carol has intercepted 3 cryptograms y_1, y_2, y_3 sent by Alice to 3 different users whose public keys are n_1, n_2, n_3, and the public exponent $e = 3$. What does Carol need to do to reconstruct the message m?

7. Show that RSA is not IND–CPA. The IND–CPA game is defined as follows

   (a) The challenger generates a new key pair PK, SK and publishes PK to the adversary, the challenger retains SK.

   (b) The adversary may perform a polynomially bounded number of calls to the encryption oracle or other operations.

   (c) Eventually, the adversary submits two distinct plaintexts M_0 and M_1 to the challenger.

   (d) The chellenger selects a bit $b \in \{0, 1\}$ uniformly at random, and sends the challenge ciphertext $C = E(PK, M_b)$ back to the adversary.

   (e) The adversary is free to perform any number of additional computations.

   (f) Finally, the adversary outputs a guess for the value b.

   A cryptosystem is said to be IND-CPA if that every probabilistic polynomial time adversary has only a negligible advantage over random guessing.

8. Show that RSA is not IND-CCA2. The IND-CCA2 game is defined as follows.

   (a) The challenger generates a new key pair PK, SK and publishes PK to the adversary, the challenger retains SK.

   (b) The adversary may perform any number calls to the encryption or decryption oracles, or other operations.

   (c) Eventually, the adversary submits two distinct chosen plaintexts M_0 and M_1 to the challenger.

   (d) The challenger selects a bit $b \in \{0, 1\}$ uniformly at random, and sends the challenge ciphertext $C = E(PK, M_b)$ back to the adversary.

   (e) The adversary is free to perform any number of additional computations, calls to the encryption and decryption oracles, but may not submit the challenge ciphertext C to the decryption oracle.

   (f) Finally, the adversary outputs a guess for the value b.

   Use the property properties of RSA, which is homomorphic w.r.t. multiplication, meaning that

$$\begin{cases} C_1 = m_1^e \bmod n \\ C_2 = m_2^e \bmod n \end{cases} \Longrightarrow C_1 \cdot C_2 = m_1^e \cdot m_2^e \bmod n = (m_1 m_2)^e \bmod n \ .$$