Exercise 1

The *ROT-13* cipher is a simple shift cipher (Caesar) with fixed key $k = 13$.

1. Give a formal definition of the cipher and show that encryption and decryption functions coincide (for the English alphabet);
2. Consider a variant in which the i-the letter is encrypted under key $k = 13 + i \mod 26$ and prove that this variant is not perfect.
3. How shall we pick i to have a perfect cipher? Discuss and prove that the proposed variant is perfect.

Exercise 2

It is proposed the following e-voting protocol:

$$A \rightarrow B : E_{PK_B}(Vote), Sign_{SK_C}(E_{PK_B}(Vote))$$

where *Vote* is A's vote, picked from a list of n candidates, PK_B is the public key of the 'ballot' server ('urna' in Italian) and SK_C is the private key of a centralized server checking that each user votes only once. We assume that this part preventing double-voting is implemented correctly.
The question is whether the above protocol guarantees vote confidentiality or not. In particular the decision is to use 2048-bits RSA with no padding to implement encryption E_{PK_B}. Discuss and show potential attacks and fixes.

Exercise 3

Consider the following mutual authentication protocol

$$\begin{array}{lcll} B & \rightarrow & A & : E_K(N_B, B) \\ A & \rightarrow & B & : E_K(N_B, B), N_A \\ B & \rightarrow & A & : E_K(N_A, A) \end{array}$$

where N_A and N_B are nonces, K is a long-term key shared between Alice and Bob.
The protocol is trivially broken. Show the problem a propose a (minimal) fix.

Exercise 4

To encourage users adopting secure passwords it is proposed to let the system pseudo-randomly generate them. In order to make them easy to memorize the length is fixed to 6 characters and symbols are only limited to upper-case English characters. Passwords are stored as SHA-256 salted hashes.
Discuss the security of the system in case an attacker gets access to the password (hashed) file.

Exercise 5 (optional, gives extra score)

What is the RSA multiplicative property? Suppose we use RSA with no hashing to sign messages: $Sign_{SK_A}(M) = E_{SK_A}(M)$. Show how the multiplicative property can be exploited to forge signatures.