Exam. Duration: Three hours.

The exam contains 4 questions. Answer all of them.

A single formula sheet of size A4 is allowed. Any other study material is forbidden during the exam. In particular, cell-phones are not allowed. Submit your formula sheet with the exam.

Devote the first 10 minutes for reading and understanding all the questions.

Make your answers as short and as clear as possible. Explain all your answers.

Allocate in advance two pages in your notebook for each question, according to the order of the questions.

Write in an orderly and clean manner, with clear handwriting. Unclear answers will not be checked.

Good Luck!
Question 1 (25 points)

a. In Rabin’s variant for RSA encryption, the public key is e=2. Show that an attacker who can decrypt messages encrypted by Rabin’s variant can find the prime factors of n.

b. In class we saw an implementation of OT which is based on the extraction of square modular roots. Describe the scheme and show that it is secure both against a cheating sender and a cheating receiver.

Reminder:
In OT the sender has a secret S.
At the end of the protocol the receiver learns S with probability 0.5, and does not learn any information about it with probability 0.5.
The receiver knows if he was able to learn S or not.
The sender does not know if the receiver learnt S or not.

Question 2 (25 points)

In this question we assume that bank branches contact the bank’s computer center using encryption in the CBC mode of operation, and setting the initial value IV to 0. Moreover, all communication between the branches and the computer center start with the string "BANKTRANSMISSION".

We know that all the money transfer messages between branches are encrypted in the above method using DES as the underlying block cipher.

For security reasons, each money transfer message is encrypted under a new key. In this question we will show that frequent changing of the encryption keys weakens this particular encryption.

Show how to transfer $100,000,000 to your account, by an attack with complexity of at most $2^{35}$. Describe your attack in detail and analyze its complexity.

Note: you can assume that many money transfer messages are issued every day, but your attack can only change one message, which will cause the money to be transferred to your account. In addition, your attack may not cause a long delay in transferring the messages, since then the bank’s suspicions may arise.

Reminder for the CBC mode of operation:
Question 3 (25 points)

A cipher is called closed under composition, if for every two keys $K_1$, $K_2$ (not necessarily distinct) there exists a key $K_3$, such that:

$$E_{K_1}(E_{K_2}(P)) = E_{K_3}(P) \quad \text{for all } P.$$

For example, the Caesar cipher is closed under composition, since encrypting under a key $K_1$, and encrypting the result under a key $K_2$, is equivalent to encrypting under a key $K_3=(K_1+K_2) \mod 26$. Another example is a substitution cipher.

For the purpose of this question we assume that $E$ is DES, and that it is closed under composition (In practice DES is not closed under composition).

a. Show that there exists a key $I$ for which for all $P$ it holds that:

$$E_I(P)=P.$$

b. Show that for every key $K$ there exists an inverse key $K'$, such that for all $P$ it holds that:

$$E_{K'}(E_K(P))=P.$$

c. Show that for every key $K_3$ and every key $K_2$, there exists a key $K_1$, such that:

$$E_{K_1}(E_{K_2}(P)) = E_{K_3}(P) \quad \text{for all } P.$$

d. Show that it is possible to break $E$ (with high probability) with complexity not exceeding $2^{35}$ encryptions/decryptions, given a plaintext $P$ and its ciphertext $C$ under the key $K$. What is the complexity of the attack you described?

Instruction: Observe the key in use as a composition of two other keys, and given $P$ and $C$, show how it is possible to encrypt and decrypt every message (if needed the attacker can require several plaintexts and their encryption).
Question 4 (25 Points)
In the home exercise you saw an identification protocol which is based on Schnorr’s signature scheme, and you showed that it is a ZK protocol.

A reminder for the identification protocol:
Let \( p \) be a large prime of size 512 bits, let \( q \) be a prime of size 140 bits which divides \( p-1 \), and let \( \alpha \in \mathbb{Z}_p^* \) with order \( q \).
The prover has a secret key \( s \in \mathbb{Z}_q^* \), and the public key is \( v = \alpha^s \mod p \).
The identification is done by repeating \( t \) times:
1. The prover randomly chooses \( r \in \mathbb{Z}_q^* \).
2. The prover computes \( x = \alpha^r \mod p \) and sends \( X \) to the verifier.
3. The verifier chooses \( e \) in the range \([1, \log p]\) and sends it to the prover.
4. The prover computes \( y = r + se \mod q \) and sends \( y \) to the verifier.
5. The verifier verifies that \( x = \alpha^e v^r \mod p \).

The simulator for the protocol:
Given a public key \( v \) and a verifier algorithm \( V^* \), do until \( t \) tuples are obtained:
1. Choose \( e' \in [1, \log p] \) and \( y \in \mathbb{Z}_q^* \).
2. Compute \( x = \alpha^{e'} v^r \mod p \), and “send” \( X \) to \( V^* \).
3. Get a challenge \( e' \) from \( V^* \).
4. If \( e = e' \) “send” \( y \) to \( V^* \), and add \((x, e, y)\) to the transcript.
   Otherwise, reset \( V^* \) to its condition in the beginning of the current iteration.

a. Explain how we can efficiently find \( p, q \), according to the requirements of the protocol.
b. How many elements of order \( q \) are there in \( \mathbb{Z}_p^* \)? How can we efficiently find such an element?
   Note: the factorization of \( p-1 \) to its prime factors in not necessarily known.

In this question we deal with a case where there are two secret keys \( s_1, s_2 \). The prover knows at least one of them. The prover’s goal is to identify himself to the verifier as knowing one of the two secret keys, without the verifier learning which. There are two public keys \( v_1, v_2 \), and all the other parameters are the same as in the standard case.
The following protocol is suggested, assuming the prover knows \( s_1 \), but not \( s_2 \).
Repeat \( t \) times:
1. The prover chooses \( r \in \mathbb{Z}_q \) and \( e_1, e_2 \in \mathbb{Z}_q \).
2. The prover computes \( x_1 = \alpha^r \mod p, x_2 = \alpha^{s_1} v_2 e_2 \mod p \) and sends them to the verifier.
3. The verifier chooses \( e \) in the range \([1, \log p]\) and sends it to the prover.
4. The prover computes \( e_1 = e \oplus e_2 \).
5. The prover computes \( y_1 = r + s_1 e_1 \mod q \) and sends \( y_1, y_2, e_1, e_2 \) to the verifier.

c. How does the verifier verify the answers of the prover?
d. Show that the given protocol is an interactive proof.
e. Is it a ZK protocol? If you answer yes – prove it. If no, describe what the verifier is able to learn.