Exam Duration: Three hours.

The exam contains 4 questions. Answer all of them.

A single formula sheet provided with the exam is allowed. Any other study material is forbidden during 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. Use as efficient algorithms as you can.

Allocate in advance two pages 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)**

In this question we discuss finding square roots modulo $p^2$, where $p$ is an odd prime.

Let $p$ be an odd prime, and let $a$ be an integer, $0 \leq a < p^2$. We want to determine whether $a$ is a quadratic residue modulo $p^2$.

a. What is the Jacobi symbol of $a$ over $p^2$?

Now, $a$ is a quadratic residue modulo $p^2$, i.e., there exists $y$ s.t. $a \equiv y^2 \pmod{p^2}$. Let $s, t$ be integers, $0 \leq s, t < p$, such that $y \equiv sp + t \pmod{p^2}$, and we would like to find $y$.

b. Given $a$, show how to find $t$, as above.

c. Given $a$ and $t$, show how to find $s$, and from both, how to find $y$.

d. Let $b$ be a quadratic residue modulo $p^3$. Show how to implement a similar technique to find a square root of $b$ modulo $p^3$.

Let us consider again the question whether $a$ is a square root modulo $p^2$:

e. How many of the elements in $Z_p$ are quadratic residues? How many of the elements in $Z_{p^2}$ are quadratic residues? Prove.

f. Under which conditions $x \in Z_{p^2}$ is a quadratic residue modulo $p^2$? (The desired answer is not to restate the definition, namely $x$ has a root modulo $p^2$.)
Question 2 (25 points)

In the Merkle-Damgard scheme (See sketch below) a message of arbitrary length is hashed by dividing it to blocks of fixed length: $M_1, M_2, \ldots, M_l$. The compression function $h_i = C(h_{i-1}, M_i)$ compresses the previous chaining value $h_{i-1}$ and the current message block $M_i$, and generates a new chaining value $h_i$. The value $h_0 = IV$ is fixed and known to everyone, and the resulting hash value is the last chaining value $h_l$, as seen in the figure.

In SHA-1 the compression function follows the Davis-Meyer (DM) construction:

$$h_i = C(h_{i-1}, M_i) = E_M(h_{i-1}) + h_{i-1},$$

where $E$ behaves like a block cipher. The IV, the chaining values $(h_i)$, and the hash value are all 160 bit long. The message blocks $M_i$ are 512 bit long.

A. Explain why the Davis-Meyer structure is necessary (adding $h_{i-1}$ at the end of the compression function). I.e., show a possible attack which is possible if the compression function is of the form: $h_i = C(h_{i-1}, M_i) = E_M(h_{i-1})$, but is not possible with the DM addition.

B. Tamar claimed that it was not necessary to perform the DM addition in every single block, and that it was enough to perform it once on all the blocks together. I.e., hashing will be done by:

$$h_i = \begin{cases} E_M(h_{i-1}) & 1 \leq i \leq l-1 \\ E_M(h_{i-1}) + h_0 & i = l \end{cases}$$

Does the attack from A still work? Is the new method vulnerable to other attacks on hash functions (which DM protects against)?

C. Ronen claimed that if we are going to perform the DM protection only once, we should do it on the last block. I.e., hashing will be done by:

$$h_i = \begin{cases} E_M(h_{i-1}) & 1 \leq i \leq l-1 \\ E_M(h_{i-1}) + h_{i-1} & i = l \end{cases}$$

Does the attack from A still work? Is the new method vulnerable to other attacks on hash functions (which DM protects against)?

D. Yafit claimed that in order to increase the mixing between the blocks of the message we should perform the hashing in the following way:

$$h_i = \begin{cases} E_M(h_{i-1}) + h_{i-1} & \text{if } i \text{ is odd} \\ E_M(h_{i-1}) + h_{i-2} & \text{if } i \text{ is even} \end{cases}$$

Does the attack from A still work? Is the new method vulnerable to other attacks on hash functions (which DM protects against)?
Question 3 (25 points)

Let DES' be a variant of DES which does not use the key scheduling algorithm: the key is 48 bits long, and in all rounds the exactly the same key is used (without shifting). The rest of the DES structure remains unchanged.

We will show a chosen plaintext attack on this variant of DES: choose a random 32 bits string \( x \); \( t \) random 32 bits strings denoted \( \alpha_i \); and \( t \) random 32 bits strings denoted \( \beta_i \).

We require \( t \) encryptions of the plaintexts \( P_i = (\alpha_i, x) \), and \( t \) encryptions of the plaintexts \( P_j^* = (x, \beta_j) \). The cyphertexts will be denoted \( C_i = (C_{Li}, C_{Ri}) \) and \( C_j^* = (C_{Li}^*, C_{Ri}^*) \), respectively.

a. What condition should hold, so that the input of the second round in the encryption of \( P_i = (\alpha_i, x) \) (64 bits) will be equal to the input of the first round of encryption of \( P_j^* = (x, \beta_j) \)? Explain.

b. For which value of \( t \), we expect to find a pair \( P_i, P_j^* \) satisfying the conditions of section a. with probability greater than \( 1/2 \)? Explain.

c. Given cyphertexts \( C_i = (C_{Li}, C_{Ri}) \), \( C_j^* = (C_{Li}^*, C_{Rj}^*) \) for which the equality from section a. holds, prove that the equality \( C_{Rj}^* = C_{Li} \) holds as well.

d. For the \( t \) value from section b., what is the expected number of pairs \( C_i, C_j^* \) for which the equality from section c. holds while the equality from section a. does not hold? Explain.

e. What other condition connects the values \( C_i, C_j^* \)? Explain.

f. Using the given \( 2t \) cyphertexts, show how to find the key using the mentioned equalities, with probability greater than \( 1/5 \) and complexity \( 2^{20} \) at the most.
Question 4 (25 Points)

Prover P knows the prime knows the prime factorization \( n = pq \), where \( p \) and \( q \) are two large primes. He would like to prove to V that he knows this factorization, without revealing the values of \( p \) and \( q \).

All the random choices at this question are done uniformly from the relevant domain.

Ela suggested repeating the following protocol \( t \) times:

1. V chooses a random value \( r \in \mathbb{Z}_n^* \), calculates \( x = r^2 \mod n \) and sends it to P.
2. P computes the modular square root of \( x \) and sends it back to V.
3. V verifies the answer.

After V had verified the answer is correct in all the iterations, he becomes convinced that P knows the factorization.

a. How stage 2 is performed?

b. What is the soundness of this protocol?

c. Does V study anything from the protocol?

Bella proposed the following improved protocol:

1. V chooses a random value \( r \in \mathbb{Z}_n^* \), computes \( x = r^2 \mod n \) and sends it back to P.
2. P chooses a random value \( y \in \mathbb{Z}_n^* \), computes \( z = xy^{-1} \mod n \), and sends \( y, z \) back to V.
3. V verifies that \( x = yz \), and asks P for the square root of \( y \) or the square root of \( z \).
4. P sends the square root back to V.
5. V verifies the answer.

This protocol is also repeated \( t \) times, and if V hadn’t rejected any of them, he becomes convinced that P knows the factorization.

d. Prove that this protocol is an interactive proof.

e. Is this protocol a zero-knowledge proof? If so, is this perfect zero-knowledge or computational zero-knowledge? Prove.