Exam

Exercise 1. Fully Homomorphic Encryption over natural numbers.

Let p, q be two large primes. Let $N = pq$. Let $\beta \ll p, q$ (*i.e.* β is an integer much smaller than p and q). Consider the following symmetric encryption scheme, with secret key p, and (public) evaluation key N.

> **Encryption.** To encrypt a message $m \in \{0, 1\}$, sample q' and r independently and uniformly at random in $\{0, \ldots, \beta\}$, and let $\mathrm{Enc}(m) = q'p + 2r + m$.
>
> **Decryption.** To decrypt a ciphertext c, compute $\mathrm{Dec}(c) = (c \bmod p) \bmod 2$. In that expression, "$a \bmod b$" outputs an integer in $\{0, \ldots, b-1\}$.

Question 1.1. Without doing detailed computations, show how it is possible to compute an encryption of the message $m + m'$ (addition modulo 2), given only an $\mathrm{Enc}(m)$ and $\mathrm{Enc}(m')$, under some circumstances. The computation may also use the evaluation key N, if necessary. Same question for $m \cdot m'$ (multiplication modulo 2), and $m + 1$.

Question 1.2. For fixed parameters N, β, is the scheme fully homomorphic, in the sense that one may compute an arbitrary boolean circuit homomorphically? If yes, justify the answer; if not, briefly explain how to modify the scheme to be fully homomorphic.

Question 1.3. Recall that in the standard IND-CCA security definition of an encryption scheme, the adversary has access to a decryption oracle. That is, the adversary can query an oracle, on an arbitrary input of their choice, and obtain a decryption of that input. Show that the above encryption scheme is not IND-CCA secure.

Exercise 2. Pedersen commitments.

Let $\mathbb{G}$ be a cyclic group of prime order p. Fix g and $h = g^\alpha$ two distinct generators of $\mathbb{G}$. The generators g and h are public, but not α (which is not known to anyone). We now describe Pedersen commitments. To *commit* to an element $x \in \mathbb{Z}_p$ means to sample r from $\mathbb{Z}_p$ uniformly at random, and to publish $c = g^x \cdot h^r$. To *open* the commitment c means to publish the corresponding values x and r (such that $c = g^x \cdot h^r$). A commitment scheme must be *hiding* and *binding*, in the sense defined in the next two questions.

Question 2.1. (Hiding property.) Show that for any value x, the distribution of c is indistinguishable from the uniform distribution over $\mathbb{G}$ (perfectly, statistically, or computationally).

Question 2.2. (Binding property.) Show that if two distinct openings (x, r) and (x', r') are provided for the same commitment c, then this implies solving a certain instance of the discrete logarithm problem.

Question 2.3. Briefly sketch an argument showing that the for a fixed matrix A of suitable size, the probabilistic map $x \mapsto Ax + e$ defined over short vectors x, and where e is sampled as a short vector for each new commitment, is a hiding and binding commitment scheme. (Provided the size of A and notion of shortness are chosen in an adequate way, such that certain computational problems are hard.)

From a Pedersen commitment, one can build a zero-knowledge proof showing that the prover knows the commited value x, without revealing x.

1. First, the prover samples u, v uniformly at random in $\mathbb{Z}_p$, and sends $d = g^u \cdot h^v$ to the verifier.
2. The verifier then samples e uniformly at random from $\mathbb{Z}_p$, and sends it to the prover.
3. The prover computes $s = u - xe$, and $t = v - re$ (in $\mathbb{Z}_p$), and sends s, t to the verifier.
4. The verifier accepts the proof iff $g^s \cdot h^t \cdot c^e = d$.

Question 2.4. Show that this interactive proof system is complete (if the prover does know x, the verifier will accept). Show that it satisfies special soundness (if the verifier accepts for two different values of e, and for the same d, then the prover does know x). Show that it is honest-verifier zero-knowledge (the transcript of an honest interaction can be simulated without knowing x). Does the zero-knowledge property hold perfectly, statistically, or only computationally?

Exercise 3. Post-quantum Signatures via Zero-Knowledge (and schizophrenia).

Notation. For $\vec{v} \in \mathbb{Z}_2^n$ and $i \in \{1, \ldots, n\}$, let v_i denote the i-the coordonnate of the vector $\vec{v}$. For $\vec{v}, \vec{w} \in \mathbb{Z}_2^n$, $\vec{v} + \vec{w}$ is the sum vector, where the sum is computed coordinate-wise modulo 2. We may also write $\vec{v} - \vec{w}$, which is equivalent. In this exercise, we write that a triple $(\vec{a}, \vec{b}, \vec{c})$ *encodes* a secret $\vec{s}$ iff $\vec{a} + \vec{b} + \vec{c} = \vec{s}$.

Multi-Party Computation (MPC)

Alice, Bob and Charlie share a secret $\vec{s} \in \mathbb{Z}_2^n$: Alice knows $\vec{a} \in \mathbb{Z}_2^n$, Bob knows $\vec{b} \in \mathbb{Z}_2^n$, and Charlie knows $\vec{c} \in \mathbb{Z}_2^n$, such that $\vec{a} + \vec{b} + \vec{c} = \vec{s}$. Only Alice knows $\vec{a}$, only Bob knows $\vec{b}$, only Charlie knows $\vec{c}$. The triple $(\vec{a}, \vec{b}, \vec{c})$ is sampled uniformly at random in $(\mathbb{Z}_2^n)^3$ among triples that satisfy the condition $\vec{a} + \vec{b} + \vec{c} = \vec{s}$. (Equivalently: $\vec{a}$ et $\vec{b}$ are sampled uniformly at random in $\mathbb{Z}_2^n$, and $\vec{c} = \vec{s} - \vec{a} - \vec{b}$.)

Question 3.1. Show that if Alice et Bob share their knowledge of $\vec{a}$ and $\vec{b}$ with each other, they still know nothing about $\vec{s}$, in the sense that $\vec{s}$ remains uniform from their point of view.

Alice, Bob and Charlie now wish to share a secret that encodes the sum $s_i + s_j$ of two bits of $\vec{s}$. Towards that end, Alice computes $a' = a_i + a_j$, Bob computes $b' = b_i + b_j$, and Charlie computes $c' = c_i + c_j$.

Question 3.2. Show that (a', b', c') is uniform among triples (x, y, z) that encode $s_i + s_j$.
Hint : equivalently, show that x and y are uniform and independent, and $z = s_i + s_j - x - y$.

In that manner, Alice, Bob and Charlie are able to "compute" a sum of two bits of the secret $\vec{s}$, in the sense that they now share a new secret that encodes the sum of the two bits. Alice, Bob and Charlie now wish to "compute" a *product* $s_i s_j$ of two bits of the secret, in the same sense. Towards that end, Alice, Bob and Charlie sample independently and uniformly at random one bit each, denoted respectively r_a, r_b, r_c. Alice sends (r_a, a_i, a_j) to Charlie, Charlie sends (r_c, c_i, c_j) to Bob, Bob sends (r_b, b_i, b_j) to Alice. Alice computes $a' = a_i a_j + a_i b_j + a_j b_i + r_a - r_b$. Bob computes $b' = b_i b_j + b_i c_j + b_j c_i + r_b - r_c$. Charlie computes $c' = c_i c_j + c_i a_j + c_j a_i + r_c - r_a$.

Question 3.3. Show that (a', b', c') is once again uniformly random among triples that encode $s_i s_j$. What happens if we remove the r_i's?

Question 3.4. Propose a way to compute the negation of a secret bit s_i, in the same sense that we have computed addition and product in the previous questions.

Question 3.5. Propose a protocol that allows Alice, Bob et Charlie to compute $F(\vec{s})$ for an arbitrary map $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$. The map F is given in the form of a boolean circuit, publicly known to all parties, and composed of addition, multiplication and negation gates. At the outcome of the computation, Alice, Bob and Charlie must know the result $F(\vec{s})$ of the computation, but they must not learn anything about $\vec{s}$, aside from $F(\vec{s})$. No proof is required, but bonus points if you can find a way to formally express the property stated in that last sentence (hint: you can draw inspiration from the way "zero-knowledge" is defined, using a simulator).

Zero-Knowledge from MPC

Notation. Let $\mathsf{H} : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ denote a hash function (with a fixed input length). For a uniform element $\vec{x} \in \mathbb{Z}_2^n$, was assume that given $\vec{y} = \mathsf{H}(\vec{x})$, and only $\vec{y}$, finding a preimage of $\vec{y}$ is a hard problem (*i.e.* finding $\vec{x'} \in \mathbb{Z}_2^n$, not necessarily distinct from $\vec{x}$, such that $\mathsf{H}(\vec{x'}) = \vec{y}$, is hard). In the remainder, we use a *commitment scheme*. This can be done as in the previous exercise, but more simply, one can also do the following. To *commit* to a value x means to sample r uniformly at random among bitstrings of some fixed (sufficiently large) length, and to send $\mathsf{H}(x \parallel r)$, where $\parallel$ denotes concatenation. To *open* the commitment means to reveal x and r.

Sylvie samples $\vec{s} \in \mathbb{Z}_2^n$ uniformly at random, and publishes $\vec{y} = \mathsf{H}(\vec{s})$. Sylvie wishes to prove in zero-knowledge thta she knows a preimage of $\vec{y}$.

Question 3.6. Propose a generic way to achieve that goal, using a technique from the course. (Details are not required, the idea can be sketched in a couple lines.)

However, we do not want to use that generic approach. Among other issues, it would not be quantum-resistant. Instead, Sylvie will prove her knowledge of a preimage of $\vec{y}$ using the following protocol.

(a) Sylvie shares a secret $\vec{s}$ among three virtual entities Alice, Bob and Charlie, by sampling $\vec{a}$, $\vec{b}$, $\vec{c}$ such that $\vec{a} + \vec{b} + \vec{c} = \vec{s}$, exactly like in the previous section. Note that Alice, Bob and Charlie only exist in Sylvie's head. Sylvie then computes $\mathsf{H}(\vec{s})$ following the same multi-party computation protocol as in the previous section, as if Alice, Charlie and Bob were really three distinct people. (For that purpose, H is viewed as a circuit.) Finally, Sylvie sends to Thomas the following information: a commitment to the secret $\vec{a}$

of Alice, a commitment to every value computed at the output of a logic gate by Alice (the values a' of questions 2 and 3), as well as the random bits r_a used in multiplication gates (question 3) ; likewise for Bob and Charlie.[1] Sylvie also sends the values $\vec{a}_f$, $\vec{b}_f$, $\vec{c}_f$, obtained by Alice, Charlie and Bob at the outcome of the computation (and such that $\mathsf{H}(\vec{s}) = \vec{a}_f + \vec{b}_f + \vec{c}_f$).

(b) Thomas chooses one person among Alice, Bob and Charlie, et and sends that choice to Sylvie.

(c) Sylvie opens all commitments related to the two people *not* chosen by Thomas.

(d) Thomas accepts the proof if all the computations he can check based on the information he has received are correct. (In more detail: he checks that: (1) the values revealed by Alice in step (c) do match the commtiments from step (a); (2) each logic gate computed by Alice, Charlie, and Bob is computed correctly, whenever Thomas knows the inputs of the gate ; (3) the final outcome of the computation is equal to $\vec{a}_f$, $\vec{b}_f$, $\vec{c}_f$, for the two virtual parties chosen in step (b); (4) $\vec{y} = \vec{a}_f + \vec{b}_f + \vec{c}_f$.)

Question 3.7. Assume that Thomas chooses Alice in step (b). Show that Thomas is able to check all computations performed by one of the other virtual parties (which one?).

Question 3.8. Suppose that Sylvie does not actually know a preimage of y. Propose a strategy for Sylvie such that Thomas will accept the proof, with probability (at least) $1/3$. Conversely, show that if Sylvie can produce a proof that will be accepted with probability 1 (relative to Thomas' choice in step (b)), then she must know a preimage. What does that imply about the zero-knowlege proof?

Question 3.9. Show that the previous protocol is *zero-knowledge*.

Question 3.10. The zero-knowledge proof in this exercise is a sigma protocol; recall that such protocols can be converted into a signature scheme via the Fiat-Shamir transform. Provide an upper bound of the signature size, as a function of the number of gates of the circuit computing H, and other parameters of the scheme (such as n, and the length of a commitment). Bonus points: what can be said about the quantum resistance of the resulting signature scheme?

[1] Note that Sylvie does not commit to the values (r_c, c_i, c_j) *received* by Alice for a multiplication gate (question 3); she only commits to the output value a'.