

Privacy Enhancing Technologies FS2026

Exercise Sheet 4 (version 1)

Florian Tramèr

Problem 1: Conceptual Questions. For each of the following statements, say whether it is TRUE or FALSE. Write at most one sentence to justify your answer.

In a world where $P = NP$ (in the standard model, with no random oracle):

- (a) the PLONK Poly-IOP is sound
- (b) secure single-server PIR schemes exist
- (c) secure two-server PIR schemes exist
- (d) all languages in NP have a SNARG

Problem 2: Some Polynomial IOPs. In all of the following problems, you should assume that the prover can send the verifier polynomial oracles for polynomials over $\mathbb{F}[X]$ with a specific degree bound k . (that is, if the prover creates an oracle for some polynomial f of degree k , then the verifier can evaluate the polynomial at arbitrary points, *and* be assured that the polynomial is of degree $\leq k$.)

- (a) Let f be a polynomial of degree d over $\mathbb{F}$. The prover wants to convince the verifier that f has no coefficients of degree less than i for some $i < d$. That is, $f(X) = a_i X^i + a_{i+1} X^{i+1} + \dots + a_d X^d$. Give a polynomial IOP for this problem with perfect completeness and soundness error $d/|\mathbb{F}|$.

Hint: Write $f(X) = X^i \cdot g(X)$ where $g(X)$ is a polynomial of degree $d - i$.

- (b) Let f be a polynomial of degree d over $\mathbb{F}$. The prover wants to convince the verifier that the i -th coefficient of f is β , i.e., $f(X) = a_0 + a_1 X + \dots + \beta X^i + \dots + a_d X^d$. Give a polynomial IOP for this problem with perfect completeness and soundness error $d/|\mathbb{F}|$.

Hint: Split the polynomial f into two polynomials around the i -th coefficient, and use a similar approach to the previous problem.

- (c) The prover now has two polynomials $f = a_0 + a_1 X + \dots + a_d X^d$ and $g = b_0 + b_1 X + \dots + b_d X^d$ of degree d over $\mathbb{F}$. They want to convince the verifier that $\langle \mathbf{f}, \mathbf{g}^{rev} \rangle = z$, where $\mathbf{f} = (a_0, a_1, \dots, a_d)$ and $\mathbf{g}^{rev} = (b_d, b_{d-1}, \dots, b_0)$. That is, the inner product between the coefficients of f and the coefficients of g in reverse order is equal to z .

Give a polynomial IOP for this problem with perfect completeness and soundness error $4d/|\mathbb{F}|$. You should assume that the prover can send the verifier polynomial oracles for polynomials over $\mathbb{F}[X]$ with a specific degree bound $k \leq 2d$.

Hint: Let $h(X) = f(X) \cdot g(X)$ be the product of the two polynomials, of degree $2d$. What is the d -th coefficient of $h(X)$?

Problem 3: Polynomial Commitments. To complete the PLONK SNARG we saw in the lecture, we still need a secure polynomial commitment scheme! The one you will build here

is essentially a ZeroTest performed “in the exponent” of a group.

Let $\mathbb{G}$ be a group of prime order p and $g \in \mathbb{G}$ be a group generator. Our scheme’s setup produces the following public parameters:

$$\{g, g^\alpha, g^{\alpha^2}, \dots, g^{\alpha^d}\},$$

where $d < p$ is a bound on the degree of the polynomial we can commit to, and $\alpha \in \mathbb{Z}_p^*$ is a random element that is *unknown* to the prover and verifier.

- (a) To commit to a polynomial $f \in \mathbb{F}[X]^{\leq d}$ (where $\mathbb{F} = \mathbb{Z}_p$), the prover sends $c = g^{f(\alpha)}$ to the verifier. Show how the prover can compute this value.
- (b) Is this commitment statistically or computationally binding? Justify your answer.
- (c) To open the commitment at some point $f(z) = y$, we will use a ZeroTest! Describe the quotient polynomial $q(X)$.
- (d) Assume the existence of an efficient algorithm $\text{VerifProd}(g^a, g^b, g^c) \rightarrow \{0, 1\}$ that outputs 1 if and only if $c = a \cdot b$ for $a, b, c \in \mathbb{Z}_p^*$.¹ Describe a verification algorithm that checks that $f(z) = y$, given $g^{f(\alpha)}$ and $g^{q(\alpha)}$ (with negligible soundness error).
- (e) Show that if the prover can compute the value $g^{\frac{x}{\alpha-z}}$ given any g^x , then they can break evaluation binding at point z .

Problem 4: PIR from FHE.

- (a) Build a boolean circuit C_{EQ} with NOT, OR and AND gates such that on input bits $a_1, \dots, a_n, b_1, \dots, b_n \in \{0, 1\}^{2n}$, the circuit outputs 1 if and only if $a_i = b_i$ for all $i \in [n]$, and outputs 0 otherwise.
- (b) Given a secret-key FHE scheme that supports encryptions, NOTs, ANDs and ORs over $\mathbb{Z}_2$, build a single-server one-round PIR scheme (for a database of n bits) in which the client uploads $O(\log n)$ ciphertexts to the server and the server responds with a single ciphertext to the client.
- (c) Suppose that FHE ciphertexts produced by Enc have some “noise” of magnitude 2. For ciphertexts c_1, c_2 with noise magnitudes $\varepsilon_1, \varepsilon_2$ respectively, the noise magnitude of $\text{AND}(c_1, c_2)$ is $\varepsilon_1 \cdot \varepsilon_2$, the noise magnitude of $\text{OR}(c_1, c_2)$ is $\varepsilon_1 + \varepsilon_2$, and the noise magnitude of $\text{NOT}(c_1)$ is ε_1 .

What is the noise magnitude of the ciphertext produced by the PIR scheme you built in the previous part?

¹You may have noticed that such an algorithm solves the DDH problem! It turns out that there exist groups where we conjecture that discrete-log is hard, but the DDH problem is easy. Such “gap groups” are extremely useful for cryptographic constructions like the one in this problem. The only known examples are groups built from elliptic curves that admit an efficient *pairing* operation (see [here](#) if you are curious).