# Week 9 Exercises (ECE 598 DA)

**Exercise (Perfect security of one-time-pad sharing):** Consider the 2-out-of-2 additive secret sharing scheme (the one-time pad XOR scheme) for a 1-bit secret $m \in \{0, 1\}$. The scheme gives one share $s_0 = r$ (a random bit) to participant 1, and one share $s_1 = m \oplus r$ to participant 2. Prove that this scheme is perfectly secure, i.e. that any one share is independent of the secret. In particular, show that $\mathbb{P}[s_0 = x \mid m = 0] = \mathbb{P}[s_0 = x \mid m = 1]$ for $x \in \{0, 1\}$, and similarly for $s_1$.

**Exercise (Perfect secrecy of Shamir's scheme):** Let $2 \leq k \leq n$. In Shamir's $(k, n)$ threshold scheme over a field $\mathbb{F}_q$ (with $q$ large), prove that any set of $t < k$ shares provides no information about the secret $S$. Specifically, fix any $t$ distinct participants and suppose they collude, possessing shares $yi_1, \ldots, y_{i_t}$ corresponding to points $(x_{i_j}, y_{i_j} = P(x_{i_j}))$ on the sharing polynomial $P(x)$. Show that for any candidate secret value $s \in \mathbb{F}_q$, there exists a polynomial $\widetilde{P}(x)$ of degree $< k$ such that $\widetilde{P}(0) = s$ and $\widetilde{P}(x_{i_j}) = y_{i_j}$ for all the $t$ shares they hold. Conclude that all secrets $s$ are equally likely given these $t$ shares. (Hint: Use the fact that there are $k$ unknown coefficients in a degree $k - 1$ polynomial, and you have only $t < k$ constraints so far.)

**Exercise (Share size vs. secret size):** Suppose we tried to design a perfectly secure $(2, 2)$ secret sharing scheme for a secret $m$ that is a uniformly random 2-bit value (so $m \in \{0, 1, 2, 3\}$ with equal probability). Show that in any such scheme, each share must take at least 2 bits to encode (so the share space must have size at least 4). In other words, it is impossible to perfectly share a 2-bit secret using shares that are only 1 bit long. (This is a specific case of a general fact: in any perfect secret sharing scheme, each share's entropy must be at least the entropy of the secret.)

**Exercise (Simulating a Verifier's View in Linear Protocol):** In the two-verifier linear sum protocol from the example, construct an explicit simulator for verifier $V_1$'s view and prove that the simulation is perfect (identical distribution to real).

**Exercise (Verifying a Multiplication in Two-Party ZK):** Suppose $P$ wants to prove to $V_0, V_1$ that three secret-shared values $a, b, c$ (with $a = [a]_0 + [a]_1$, etc.) satisfy $c = a \cdot b$. Outline a protocol to do this in a statistically sound and zero-knowledge way.

**Exercise (Collusion and Soundness Break):** In the above multiplication protocol, suppose verifier $V_1$ is malicious and colludes with the prover. Describe how they might cheat to make $V_0$ accept a false statement (i.e., $c \neq ab$) without being detected.

**Exercise (Single vs. Multi-Verifier):** The Power of Two Verifiers. Consider the **NP**-complete problem 3-Coloring: given a graph, prove it is 3-colorable without revealing the coloring.

- Why is the classic ZK protocol for graph 3-coloring (which uses a single verifier and commitments) not statistical zero-knowledge?
- Sketch how a two-verifier statistical ZK protocol for 3-coloring might avoid the need for one-way functions or commitments.

**Exercise (Multi-Verifier Extension):** How might the two-verifier protocols discussed be extended to $n$ verifiers? In particular, describe the secret-sharing and trust assumptions for $n$ verifiers, and comment on how the simulation condition generalizes.