CS 355: Topics in Cryptography **Spring 2023**

Problem Set 3

Due: 10pm, Monday, 15 May 2023 (submit via Gradescope)

Instructions: You **must** typeset your solution in LaTeX using the provided template:

`https://crypto.stanford.edu/cs355/23sp/homework.tex`

Submission Instructions: You must submit your problem set via Gradescope. Please use course code **XV5WJ4** to sign up. Note that Gradescope requires that the solution to each problem starts on a **new page**.

Bugs: We make mistakes! If it looks like there might be a mistake in the statement of a problem, please ask a clarifying question on Ed.

Problem 1: Conceptual Questions [6 points]. For each of the following statements, say whether it is TRUE or FALSE. Write *at most one sentence* to justify your answer.

(a) Let $\langle P, V \rangle$ be an interactive proof system for a language $\mathcal{L}$ with a *randomized* verifier. If $\langle P, V \rangle$ satisfies perfect completeness (i.e., completeness holds with probability 1) and perfect soundness (i.e., soundness holds with probability 1), then there is an interactive proof system for $\mathcal{L}$ with a *deterministic* verifier.

(b) Let $\langle P, V \rangle$ be a zero-knowledge interactive protocol for some language. The protocol has perfect completeness and soundness error 1/3. Which of the following are true:

i A malicious verifier interacting with an honest prover will always accept a true statement.

ii An honest verifier interacting with a malicious prover will "learn nothing" besides the statements validity.

(c) If an interactive proof $\langle P, V \rangle$ for an NP language $\mathcal{L}$ is a proof of knowledge with negligible knowledge error, then $\langle P, V \rangle$ has negligible soundness error (i.e., a malicious prover can convince an honest verifier of a false statement with at most negligible probability).

(d) Consider a modified version of Schnorr's signature in which the signing nonce r is computed as $r \leftarrow H(m)$, where $H : \{0,1\}^* \to \mathbb{Z}_q$ is a hash function, m is the message to be signed, and q is the order of the group used for the signature scheme. This deterministic version of Schnorr's signature scheme is secure.

(e) The Fiat-Shamir heuristic (as discussed in class) is a way to construct non-interactive zero-knowledge proofs *without* needing to rely on random oracles.

(f) Consider a hash function $H : \mathcal{W} \to \mathcal{X}$, and the NP-relation $\mathcal{R}_n$ for knowledge of n pre-images of H. Formally, $\mathcal{R}_n$ has instance space $\mathcal{X}^n$, witness space $\mathcal{W}^n$, and is defined by $\{(x \in \mathcal{X}^n, w \in \mathcal{W}^n) : H(w_1) = x_1 \wedge H(w_2) = x_2 \wedge \cdots \wedge H(w_n) = x_n\}$. A SNARG for $\mathcal{R}_n$ must have $o(n)$ verification time.

Problem 2: Understanding Interactive Proofs [15 points]. *(Problems from "The Foundations of Cryptography - Volume 1, Basic Techniques" by Oded Goldreich)*

(a) *The role of verifier randomness:* Let L be a language with a sound and complete interactive proof system where the verifier V is deterministic. Show that $L \in \mathsf{NP}$.

(b) *The role of prover randomness:* Let L be a language with a sound and complete interactive proof system. Show that there exists a sound and complete interactive proof system for L for which the prover P is deterministic.
[**Hint:** Use the fact that P is unbounded.]

(c) *The role of errors:* Let L be a language with a perfectly sound and complete interactive proof system, that is if $x \notin L$, the verifier *never* accepts (not even with negligible probability). Show that $L \in \mathsf{NP}$.

Problem 3: Sigma Protocol for Circuit Satisfiability [10 points]. Let circuit-SAT be the language of satisfiable Boolean circuits[1] :

$$\mathsf{circuit\text{-}SAT} = \left\{ C \colon \{0,1\}^n \to \{0,1\} \mid n \in \mathbb{N}, \exists (x_1, \ldots, x_n) \in \{0,1\}^n \text{ such that } C(x_1, \ldots, x_n) = 1 \right\}.$$

Let $\mathsf{Commit} \colon \{0,1\} \times \mathcal{R} \to \mathcal{C}$ be a perfectly-binding and computationally-hiding commitment scheme with message space $\{0,1\}$, randomness space $\mathcal{R}$, and commitment space $\mathcal{C}$. Suppose that there exist Sigma protocols $\langle P_{\text{XOR}}, V_{\text{XOR}} \rangle$ and $\langle P_{\text{AND}}, V_{\text{AND}} \rangle$ for languages $\mathcal{L}_{\text{XOR}}$ and $\mathcal{L}_{\text{AND}}$, respectively, where:

$$\mathcal{L}_{\text{XOR}} = \left\{ (c_1, c_2, c_3) \in \mathcal{C}^3 \;\middle|\; \begin{array}{c} \exists (m_1, m_2, m_3) \in \{0,1\}^3, (r_1, r_2, r_3) \in \mathcal{R}^3 \text{ such that} \\ \forall i \in \{1,2,3\} \;\; c_i = \mathsf{Commit}(m_i; r_i) \text{ and } m_1 \oplus m_2 = m_3 \end{array} \right\}$$

$$\mathcal{L}_{\text{AND}} = \left\{ (c_1, c_2, c_3) \in \mathcal{C}^3 \;\middle|\; \begin{array}{c} \exists (m_1, m_2, m_3) \in \{0,1\}^3, (r_1, r_2, r_3) \in \mathcal{R}^3 \text{ such that} \\ \forall i \in \{1,2,3\} \;\; c_i = \mathsf{Commit}(m_i; r_i) \text{ and } m_1 \wedge m_2 = m_3 \end{array} \right\}.$$

Give a Sigma protocol for circuit-SAT. In addition to describing a protocol, you will also need to show that your protocol satisfies completeness, soundness, and honest-verifier zero-knowledge. [**Hint:** When showing that your protocol is honest-verifier zero-knowledge, you may want to use a hybrid argument. One of your hybrids might rely on the commitment scheme being computationally hiding, and the other hybrid might rely on the underlying Sigma protocols being honest-verifier zero-knowledge.]

Problem 4: Polynomial Commitments: Multiple Inputs [5 pts]. The KZG polynomial commitment scheme allows for many kinds of *aggregate* proofs: a single group element that serves as a proof for multiple evaluations. This problem involves a simple example of aggregation: showing that a polynomial f evaluates to y at x_0 *and* at x_1. A PCS with *dual-input aggregation* additionally has algorithms Open2 and Check2 with syntax:

- $\mathsf{Open2}(pp, f, x_0, x_1) \to \pi$: Creates an opening proof for f's evaluation at x_0 **and** x_1.
- $\mathsf{Check2}(pp, c, x_0, x_1, y) \to \{0,1\}$: Checks that π proves $y = f(x_0)$ **and** $y = f(x_1)$.

A PCS has *dual-input aggregate correctness* and *dual-input aggregate evaluation binding* if the following hold:

[1] You can assume without loss of generality that a Boolean circuit consists of only XOR and AND gates.

- **(Perfect) Dual-Input Aggregate Correctness:** For all d, all polynomials f, and all inputs x_0 and x_1 such that $f(x_0) = f(x_1) = y$,

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(d) \\ c \leftarrow \mathsf{Commit}(pp, f) \\ \pi \leftarrow \mathsf{Open2}(pp, f, x_0, x_1) \end{array} : \mathsf{Check2}(pp, c_0, x_0, x_1, y, \pi) = 1\right] = 1$$

- **Dual-Input Aggregate Evaluation Binding:** For all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(d) \\ (c, \pi, \pi', y, y', x_0, x_1, i) \leftarrow \mathcal{A}(pp, d) \end{array} : \begin{array}{c} \mathsf{Check2}(pp, c, x_0, x_1, y, \pi) = 1 \\ \wedge \mathsf{Check}(pp, c, x_i, y', \pi') = 1 \\ \wedge\, i \in \{0,1\} \wedge y' \neq y \end{array}\right] = \mathrm{negl}(\lambda).$$

  Notice that dual-input aggregate evaluation binding asks the adversary to build a *conventional* proof that conflicts with the *aggregate* proof. This simplifies the definition considerably.

Define Open2 and Check2 for the KZG commitment scheme, such that an aggregate proof is one group element. Show that your construction has Dual-Input Aggregate Correctness and Dual-Input Aggregate Evaluation Binding, assuming t-BSDH.

Problem 5: Time Spent [1 point for answering]. How long did you spend on this problem set? This is for calibration purposes, and the response you provide will not affect your score.

Optional Feedback [0 points]. Please answer the following questions to help us design future problem sets. You do not need to answer these questions, and if you would prefer to answer anonymously, please use this form. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) What was your favorite problem on this problem set? Why?

(b) What was your least favorite problem on this problem set? Why?

(c) Do you have any other feedback for this problem set?

(d) Do you have any other feedback on the course so far?