CS 355: Topics in Cryptography **Spring 2020**

Problem Set 3

Due: May 18, 2020 at 11:59pm

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

https://crypto.stanford.edu/cs355/20sp/homework.tex

Submission Instructions: You must submit your problem set via Gradescope. 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 Piazza.

Problem 1: Conceptual Questions [10 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 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.

(b) Consider a modified version of Schnorr's signature in which the signing nonce r is computed as $r \leftarrow H(m)$, where $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$ is a hash function (modeled as a random oracle), 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.

(c) The security of the Fiat-Shamir transform implies that a sigma protocol with a random challenge and soundness 1/2 can be *directly* converted to a NIZK by replacing the challenge message with a hash, so long as the hash function is modeled as a random oracle.

(d) Recall the SNARG constructed in class from a linear PCP. If the linear PCP has soundness error ϵ, then the SNARG also has soundness error ϵ.

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 an 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 an interactive proof system. Show that there exists an 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 an interactive proof system with perfect soundness, 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] :

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

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: SNARGs in the Random Oracle Model [12 points]. In this problem, we will show how to leverage probabilistically-checkable proofs (PCPs) to construct a succinct non-interactive argument (SNARG) in the random oracle model. We will rely on the following adaptation of the famous PCP theorem:

Theorem (PCP)**.** Let $\mathcal{L}$ be an NP language. There exists two efficient algorithms $(\mathcal{P},\mathcal{V})$ defined as follows:

- The prover algorithm $\mathcal{P}$ is a deterministic algorithm that takes as input a statement $x \in \{0,1\}^n$, a witness $w \in \{0,1\}^h$ and outputs a bitstring $\pi \in \{0,1\}^m$, where $h, m = \text{poly}(n)$. We refer to π as the proof string.
- The verifier algorithm $\mathcal{V}^\pi$ is a *randomized* algorithm that takes as input a statement $x \in \{0,1\}^n$ and has oracle access to a proof string $\pi \in \{0,1\}^m$. The verifier reads $O(1)$ bits of π. The verifier chooses the bits it reads *nonadaptively* (i.e., they can depend on the statement x, but *not* on the values of any bit in π).

Moreover, $(\mathcal{P},\mathcal{V})$ satisfy the following properties:

- **Completeness:** For all $x \in \mathcal{L}$, if w is a valid witness for x, then

$$\Pr[\mathcal{V}^\pi(x) = 1\colon \pi \leftarrow \mathcal{P}(x,w)] = 1.$$

- **Soundness:** If $x \notin \mathcal{L}$, then for all $\pi \in \{0,1\}^m$,

$$\Pr[\mathcal{V}^\pi(x) = 1] \le 1/2.$$

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

(a) Let λ be a security parameter and let $H : \{0,1\}^* \to \{0,1\}^\lambda$ be a collision-resistant hash function. Use H to construct a commitment scheme $(\mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ with the following properties:

- $\mathsf{Commit}(x) \to c$: The commitment algorithm should take a message $x \in \{0,1\}^m$ and output a commitment $c \in \{0,1\}^\lambda$.
- $\mathsf{Open}(x, c, i) \to \sigma$: The open algorithm takes a message $x \in \{0,1\}^m$, a commitment $c \in \{0,1\}^\lambda$, and an index $i \in [m]$, and outputs an opening σ.
- $\mathsf{Verify}(c, i, b, \sigma) \to \{0,1\}$: The verification algorithm takes a commitment $c \in \{0,1\}^\lambda$, an index $i \in [m]$, a value $b \in \{0,1\}$, and an opening σ, and outputs a bit.

Show that your commitment scheme satisfies the following properties:

- **Completeness:** For all $x \in \{0,1\}^m$ and $i \in [m]$,

$$\Pr[\mathsf{Verify}(c, i, x_i, \sigma) = 1 : c \leftarrow \mathsf{Commit}(x); \sigma \leftarrow \mathsf{Open}(x, c, i)] = 1.$$

- **Binding:** For all efficient adversaries $\mathcal{A}$, if we set $(c, i, (b, \sigma), (b', \sigma')) \leftarrow \mathcal{A}(1^\lambda)$, then

$$\Pr[b \neq b' \text{ and } \mathsf{Verify}(c, i, b, \sigma) = 1 = \mathsf{Verify}(c, i, b', \sigma')] = \mathrm{negl}(\lambda).$$

- **Succinctness:** The commitment c output by Commit and opening σ output by Open satisfy $|c| = O(\lambda)$ and $|\sigma| = O(\lambda \log m)$.

In other words, the commitment scheme $(\mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ allows a user to succinctly commit to a long bitstring and then selectively open up a single bit of the committed string. (In this question, we do not require any hiding properties from the commitment scheme.)

(b) Let $\mathcal{L}$ be an NP language (with statements of length n). Show how to construct a 3-round succinct argument system for $\mathcal{L}$ using your commitment scheme from Part (a). Specifically, your argument system should satisfy perfect completeness, have soundness error $\mathrm{negl}(\lambda)$ against computationally-bounded provers, and the total communication complexity between the prover and the verifier should be $\mathrm{poly}(\lambda, \log n)$. In particular, the communication complexity scales *polylogarithmically* with the length of the NP statement. [**Hint:** Use the PCP theorem.]

(c) Explain how to convert your succinct argument from Part (b) into a SNARG in the random oracle model.