Privacy Enhancing Technologies **Fall 2024**

Problem Set 1 – 16/10/2024 update

Due: Fr, Oct 18 at 11:59pm CET (submit via Gradescope)

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

https://spylab.ai/teaching/pets-f24/template.tex

Submission Instructions: You must submit your problem set via Gradescope. Please use the course code provided in Moodle 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 Moodle.

Problem 1: Conceptual Questions [8 points]. For each of the following statements, say whether it is TRUE or FALSE. **Justify your answer in one sentence.**

(a) In a world where $\mathsf{P} = \mathsf{NP}$,

i. secure commitment schemes exist in the random oracle model.

ii. all languages in NP have zero-knowledge proofs.

iii. additive secret sharing schemes exist.

iv. maliciously secure 2PC (MPC for n=2 parties) protocols exist for all functions that can be computed by a circuit that has polynomial size in the input.

(b) Let Enc be a semantically-secure encryption scheme, and H be a cryptographic hash function with output size $O(\lambda)$ (modeled as a random oracle). Define $C(m,r) = \mathsf{H}(r)||\mathsf{Enc}(r,m)$. This is a secure commitment scheme.

(c) For every function $f : \mathcal{X} \times \mathcal{X} \to \{0,1\}$, there exists a secure two-party computation protocol for f, such that if the two parties' inputs are equal, the protocol reveals to each party the input of the other party.

Problem 2: Maliciously secure MPC [12 points]. We assume an authenticated broadcast channel which reliably delivers messages to all parties (i.e., when party i sends message m, all parties receive m and know that party i sent it to everyone). We further assume that no party aborts the protocol by not sending a message and that Beaver triplets have been set up in advance.

(a) Consider the MPC protocol from the lecture applied to some circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}$. Show that a malicious party can cause all benign parties to receive an incorrect answer $\hat{b} \in \{0,1\}$.

(b) As a first step towards a maliciously-secure protocol, we need a *fair* communication channel. Such a channel has the following interface:

- $\mathsf{Push}(x^{(i)})$: the i-th party pushes their message.
- $\mathsf{Pull} \to x^{(i)}$ or $\perp$: if all n parties have pushed their message and party i calls Pull, all other parties receive party i's message. If one or more parties have not pushed yet, the call has no effect.

Informally, this communication channel ensures that no party can choose their message based on the value of other messages. Show how to implement Push and Pull using tools seen in class, and argue that your construction is secure.

(c) Show that if all parties communicate over a fair channel, a malicious party can still cause all benign parties to receive an arbitrary output $\hat{b} \neq C(x^{(1)}, \ldots, x^{(n)})$.

(d) To prevent such tampering, as described in (c), modern MPC protocols *authenticate* the secret shares and check that opened values are correct. Specifically, for each secret-shared value x, there is an associated "tag"

$$t = a \cdot x\,,$$

where a is a global secret that is additively secret-shared among all parties. The value x and the tag t are always secret shared among all parties.

Given shares of $[a]$ and of values $[x], [y]$ with associated tags $[t], [u]$, show how the parties can compute authentication tags for $[x+y]$ and $[x \cdot y]$.

(e) Suppose all parties open the shares of some value $[x]$. Give a protocol that allows the parties to check that x has a correct tag. Your protocol should require all parties to send a single message to all other parties. If all parties behave honestly, noone should learn the global secret a. Prove that your scheme reveals nothing about a to honest parties, and that a cheating party (who reveals some incorrect share $\hat{x}_i = x_i + \Delta$) must know a to avoid being caught.

Problem 3: 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 $\mathbb{C}$ be a family of arithmetic circuits. 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 circuit $C \in \mathbb{C}$ and a witness $w \in \mathbb{F}^m$, and outputs a bitstring $\pi \in \{0,1\}^n$, where $n = \text{poly}(|C|)$. We refer to π as the proof string.
- The verifier algorithm $\mathcal{V}^\pi$ is a *randomized* algorithm that takes as input a circuit $C \in \mathbb{C}$ and a proof string $\pi \in \{0,1\}^n$. The verifier reads $O(1)$ bits of π. The verifier chooses the bits it reads *non-adaptively* (i.e., they can depend on the circuit C, but *not* on the values of any bit in π). You may assume that these bits are independent and uniformly random in $[n]$.

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

- **Completeness:** For all $C \in \mathbb{C}$, if $C(w) = 0$, then

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

- **Soundness:** If C is not satisfiable, then for all $\pi \in \{0,1\}^n$,

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

(a) Let λ be a security parameter and let $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ 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 $x \in \{0,1\}^n$ and output a commitment $c \in \{0,1\}^\lambda$.
- $\mathsf{Open}(x, c, i) \to \sigma$: The open algorithm takes a message $x \in \{0,1\}^n$, a commitment $c \in \{0,1\}^\lambda$, and an index $i \in [n]$, 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 $i \in [n]$, 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 $x \in \{0,1\}^n$ and $i \in [n]$,

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

- **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 $|\sigma| = O(\lambda \log n)$.

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 $\mathcal{C}$ be a family of arithmetic circuits (with circuits of size N). Show how to construct a 3-round succinct argument system for $\mathbb{C}$ 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 size of the circuit. [**Hint:** Use the PCP theorem.]

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

Problem 4: Soundness of Fiat-Shamir [4 points]. Let Σ be a Sigma protocol.

(a) Show that if Σ has soundness error $1/3$, then applying the Fiat-Shamir transform *directly* to Σ yields a non-interactive zero-knowledge proof (NIZK) that is not sound. In particular, demonstrate an efficient attack that breaks the soundness of this NIZK.

(b) Use Σ to construct a Sigma protocol Σ' that has negligible soundness error, and prove that its soundness error is negligible. Then explain how to apply the Fiat-Shamir transform to this protocol. Briefly argue why the protocol is still sound after the transformation.

Problem 5: Feedback [0 points]. Please answer the following questions to help us design future problem sets. You are not required 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) Roughly how long did you spend on this problem set?

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

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

(d) Any other feedback for this problem set?