CS 355: Topics in Cryptography **Spring 2018**

Problem Set 2

Due: April 27, 2018 at 5pm (submit via Gradescope)

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

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

Submission Instructions: You must submit your problem set via Gradescope. Please use course code **9KY4BB** to sign up. Note that Gradescope requires that the solution to each problem starts on a **new page**.

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) If there exists a PRG with 1-bit stretch, there exists a PRG with n^{800}-bit stretch (where n is the length of the PRG seed).

(b) If $\mathsf{P} = \mathsf{NP}$, then PRFs exist.

(c) The full-domain hash (FDH) signature scheme we saw in lecture builds a signature scheme from a trapdoor one-way permutation and a hash function H. The FDH scheme is secure when H is a collision-resistant hash function.

(d) 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.

(e) 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).

Problem 2: Commitments from Discrete Log [10 points]. Let $\mathbb{G}$ be a group of prime order q in which discrete log is hard. Let g and h be generators of $\mathbb{G}$. As you saw in CS 255, the Pedersen commitment scheme commits to a message $m \in \mathbb{Z}_q$ using randomness $r \in \mathbb{Z}_q$ as $\mathsf{Commit}(m; r) := g^m h^r \in \mathbb{G}$. The "public parameters" associated with the Pedersen commitment scheme is the description of the prime-order group $\mathbb{G}$ and the group elements g and h. (See CS 255 HW3 Problem 3 for a refresher on commitments.)

(a) Use $\mathbb{G}$ to construct a commitment scheme $\mathsf{Commit}_n(m_1, \ldots, m_n; r)$ that commits to a length-n vector of messages $(m_1, \ldots, m_n) \in \mathbb{Z}_q$ using randomness $r \in \mathbb{Z}_q$ with the type signature: $\mathsf{Commit}_n \colon \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{G}$. Prove that your commitment scheme is perfectly hiding and computationally binding (assuming hardness of discrete log in $\mathbb{G}$). You should specify both the public parameters of your scheme (which may be different from that of the basic Pedersen commitment scheme) as well as the description of the Commit_n function.

(b) *The original formulation of this problem was incorrect and not easily fixable, so we are removing it from the problem set.*

(c) Show that if you are given a hash function $H\colon \mathbb{Z}_q \to \mathbb{G}$ (modeled as a random oracle), the public parameters for your construction from Part (a) only needs to consist of the description of the group $\mathbb{G}$ and the description of H. Argue *informally* why your construction is secure. You do *not* need to provide a formal proof. This problem shows that getting rid of public parameters is another reason why random oracles are useful in practice!

(d) **Extra Credit [5 points].** Prove formally that your construction from Part (c) is secure in the random oracle model.

Problem 3: Sigma Protocol for Circuit Satisfiability [10 points]. For a parameter $\ell \in \mathbb{N}$, let $\mathsf{circuit\text{-}SAT}_\ell$ be the language of Boolean circuit satisfiability [1] (on ℓ-bit inputs):

$$\mathsf{circuit\text{-}SAT}_\ell = \left\{C\colon \{0,1\}^\ell \to \{0,1\} \mid \exists (x_1,\ldots,x_\ell) \in \{0,1\}^\ell \text{ such that } C(x_1,\ldots,x_\ell) = 1\right\}$$

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 $\mathcal{L}_{\text{XOR}}$ and $\mathcal{L}_{\text{AND}}$, respectively:

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

(a) Give a Sigma protocol for $\mathsf{circuit\text{-}SAT}_\ell$. 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.]

(b) **Extra Credit [5 points].** Give Sigma protocols for $\mathcal{L}_{\text{XOR}}$ and $\mathcal{L}_{\text{AND}}$ when the underlying commitment scheme is instantiated using the Pedersen commitment scheme (discussed in Problem 2).

Problem 4: Extending Oblivious Transfers [15 points]. Oblivious transfer (OT) is an important building block of many secure MPC protocols. Because OT requires public-key cryptography, implementing a large number of OTs can be very expensive in practice. In this problem, we will show how we can realize $n = \text{poly}(\lambda)$ instances of 1-out-of-2 OTs on ℓ-bit strings (where $\ell = \text{poly}(\lambda)$) using just λ instances of 1-out-of-2 OTs on λ-bit strings. Here, $\lambda \in \mathbb{N}$ is a security parameter. This means that we can essentially obtain an *arbitrary* polynomial number of OTs using a fixed number of *base OTs*.

(a) First, we show how to realize n instances of 1-out-of-2 OTs on ℓ-bit strings using λ instances of 1-out-of-2 OTs on n-bit strings (we refer to these as the *base OTs*). Consider the following protocol:

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

- Let $r \in \{0,1\}^n$ be the receiver's choice bits for the n OT instances, let $\left(m_0^{(1)}, m_1^{(1)}\right), \ldots, \left(m_0^{(n)}, m_1^{(n)}\right)$ be the sender's messages for the n OT instances.
- The receiver begins by choosing a matrix $\mathbf{M} \xleftarrow{R} \{0,1\}^{n \times \lambda}$. The sender chooses a random string $s \xleftarrow{R} \{0,1\}^{\lambda}$.
- The sender and the receiver now perform λ instances of an 1-out-of-2 OT on n-bit strings, but with their roles swapped (namely, the sender plays the role of the receiver in the base OTs, and vice versa). In the i^{th} base OT (where $i \in [\lambda]$), the sender provides s_i as its choice bit and the receiver provides $(\mathbf{M}_i, \mathbf{M}_i \oplus r)$ as its two messages, where $\mathbf{M}_i \in \{0,1\}^n$ denotes the i^{th} column of $\mathbf{M}$.
- Let $\mathbf{T} \in \{0,1\}^{n \times \lambda}$ be the matrix the sender receives from these base OTs (where the i^{th} column of $\mathbf{T}$ is the column it received from the i^{th} base OT). By construction, either $\mathbf{T}_i = \mathbf{M}_i$ or $\mathbf{T}_i = \mathbf{M}_i \oplus r$.

For $j \in [n]$, let $\mathbf{M}^{(j)}$ denote the j^{th} row of $\mathbf{M}$ and let $\mathbf{T}^{(j)}$ denote the j^{th} row of $\mathbf{T}$. Write down an expression for $\mathbf{M}^{(j)}$ in terms of $\mathbf{T}^{(j)}$ and s when $r_j = 0$ and when $r_j = 1$.

(b) Let $H\colon [n] \times \{0,1\}^{\lambda} \to \{0,1\}^{\ell}$ be a hash function (modeled as a random oracle). Suppose the sender encrypts each pair of messages $\left(m_0^{(j)}, m_1^{(j)}\right)$ by computing

$$\mathsf{ct}_0^{(j)} \leftarrow m_0^{(j)} \oplus H\left(j, k_0^{(j)}\right) \quad \text{and} \quad \mathsf{ct}_1^{(j)} \leftarrow m_1^{(j)} \oplus H\left(j, k_1^{(j)}\right),$$

where $k_0^{(j)}, k_1^{(j)} \in \{0,1\}^{\lambda}$. The sender sends each pair of encrypted messages to the receiver. Write down expressions for $k_0^{(j)}, k_1^{(j)}$ that would enable the receiver to learn $m_{r_j}^{(j)}$. [**Hint:** Recall that the receiver chooses the matrix $\mathbf{M}$ and then use the result from Part (a).]

(c) Give a brief explanation of why the receiver learns nothing about the other message $m_{1-r_j}^{(j)}$. [**Hint:** Use the fact that H is modeled as a random oracle.]

(d) Give a brief explanation of what goes wrong if we implement the hash function $H(j,k)$ as $F(k,j)$, where $F\colon \{0,1\}^{\lambda} \times [n] \to \{0,1\}^{\ell}$ is a secure PRF.

(e) Describe how you would modify the above protocol so that the base OTs are performed over λ-bit strings rather than $n = \text{poly}(\lambda)$-bit strings. [**Hint:** You will need to introduce a computational assumption.]

Problem 5: Time Spent [3 points 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?