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

Problem Set 3

Due: May 11, 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 [12 points]. For each of the following statements, say whether it is TRUE or FALSE. Write *at most one sentence* to justify your answer.

(a) 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.

(b) In the standard model (without random oracles), there exist non-interactive zero-knowledge proofs for languages in BPP.

(c) In Yao's protocol for secure two-party computation of a function $f(\cdot,\cdot)$ (as described in lecture), the two parties must exchange a number of bits that is at least as large as a Boolean circuit computing f.

(d) Say that Alice, with input $x \in \{0,1\}$, and Bob, with input $y \in \{0,1\}$, use Yao's protocol to compute $f(x,y) \in \{0,1\}$, for some function $f : \{0,1\} \times \{0,1\} \to \{0,1\}$. Then, under reasonable computational assumptions, the protocol must hide y from Alice—that is, Alice's probability of guessing Bob's bit after running the protocol is at most $1/2 + \mathrm{negl}(\lambda)$, for security parameter λ.

(e) Any pair of points $(x_1, y_1), (x_2, y_2) \in \mathbb{Z}_6^2$, where $x_1 \neq x_2$ defines a polynomial of degree at most 1 over $\mathbb{Z}_6$. (Note: "6" is not a prime.)

(f) Suppose the short integers solutions (SIS) assumption holds (for some setting of parameters). Then, pseudorandom permutations (PRPs) exist.

Problem 2: Generating Beaver Multiplication Triples [15 points]. Recall from lecture that Beaver multiplication triples enables general multiparty computation on secret-shared data. In this problem, we will explore two methods that can be used to generate Beaver multiplication triples. For simplicity, we will just consider the two-party setting and we will generate Beaver multiplication triples over the binary field $\mathbb{Z}_2$ (where addition corresponds to xor). To be precise, we first describe an "idealized process" for generating a single multiplication triple. In this "idealized process", a trusted party generates the triple and then distributes the shares of the triple to the two parties Alice and Bob.

1. The trusted party chooses $a, b \xleftarrow{R} \mathbb{Z}_2$ and computes $c = ab \in \mathbb{Z}_2$.
2. The trusted party distributes a 2-out-of-2 secret sharing of a, b, and c to Alice and Bob. Specifically, the trusted party samples $r_a, r_b, r_c \xleftarrow{R} \mathbb{Z}_2$ and gives r_a, r_b, r_c to Alice. The trusted party then computes $s_a = a \oplus r_a$, $s_b = b \oplus r_b$, and $s_c = c \oplus r_c$, and gives s_a, s_b, s_c to Bob.

By construction $[a] = (r_a, s_a)$ is an additive secret-sharing of a, $[b] = (r_b, s_b)$ is an additive secret-sharing of b, and $[c] = (r_c, s_c)$ is an additive secret-sharing of c. Moreover, $c = ab$, so $([a], [b], [c])$ is a valid Beaver multiplication triple.

We will show how Alice and Bob can generate these Beaver triples without relying on a trusted party. Throughout this problem, you may assume that Alice and Bob are "honest-but-curious" (namely, they follow the protocol exactly as described, but may try to infer additional information from the protocol transcript—this is the model that we considered in lecture).

(a) Show how Alice and Bob can generate a Beaver multiplication triple using Yao's protocol.[1] Your construction should not make any modifications to the internal details of Yao's protocol (in fact, any secure two-party computation protocol can be used here). Then, give an *informal* argument why your protocol is correct and secure. [**Hint:** To apply Yao's protocol, you will need to come up with a two-party functionality f that Alice and Bob will jointly compute. Try letting Alice's inputs to f be her shares (r_a, r_b, r_c), which she samples uniformly at random at the beginning of the protocol.]

(b) Show how Alice and Bob can use a *single invocation* of an 1-out-of-4 oblivious transfer (OT) protocol (on 1-bit messages) to generate a Beaver multiplication triple. Give an *informal* argument why your protocol is correct and secure. (In a 1-out-of-n OT, the sender has n messages $m_1, \ldots, m_n$, while the receiver has a single index $i \in [n]$. At the end of the protocol execution, the sender learns nothing while the receiver learns m_i (and nothing else). The formal definitions of sender and receiver privacy are the analogs of those presented in lecture.) [**Hint:** Try using OT to directly evaluate the functionality f you constructed from Part (a).]

(c) Let $\ell \in \mathbb{N}$ be a constant. Show how to build a 1-out-of-2^ℓ OT protocol (on 1-bit messages) using ℓ invocations of an 1-out-of-2 OT protocol (on λ-bit messages) together with a PRF $F\colon \{0,1\}^\lambda \times \{0,1\}^\ell \to \{0,1\}$. Here, $\{0,1\}^\lambda$ is the key-space of the PRF and $\{0,1\}^\ell$ is the domain of the PRF. Then, give an *informal* argument for why your protocol satisfies correctness, sender privacy, and receiver privacy. [**Hint:** Start by having the sender sample 2ℓ independent PRF keys. The sender will use these keys to blind each of its messages $m_1, \ldots, m_{2^\ell}$.]

Problem 3: Secret Sharing and Polynomials [10 points.] Let $\mathbb{F}$ be a finite field. (If it's helpful, you can think of $\mathbb{F}$ as the set of integers modulo a prime p.)

While working late into the night on your CS355 problem set, you miraculously discover a polynomial-time factoring algorithm! To safeguard your algorithm, you express it as a message $m \in \mathbb{F}$, and split it into n shares using the *additive* secret-sharing scheme we saw in lecture. You give one share s_i to each of your n friends (for $n \ll |\mathbb{F}|$) and then you spend many hours watching mind-numbing Netflix shows to erase the algorithm from your memory.

Later on, when you want to recover your secret message/algorithm m, you call your friends over to your dorm room. You ask each friend $i \in [n]$ to announce their share $s_i \in \mathbb{F}$ one at a time. Given these shares, you and your friends can all use the reconstruction algorithm to recover $m \in \mathbb{F}$.

(a) In the setting described above, each friend $i \in [n]$ announces their share s_i one at a time. Unfortunately, you are unlucky and the last friend to announce their share is not your friend, but is actually a frenemy. Show that by deviating from the specified protocol, the frenemy can broadcast a malformed

[1] You may use the variant of Yao's protocol where only one party receives output (and the other party learns nothing).

share $s'_n \neq s_n$ that will cause you to recover a message $m' \neq m \in \mathbb{F}$, for any message m' of the frenemy's choosing.

(b) One (ineffective) way to prevent the attack would be to require all of your friends to announce their shares simultaneously. Show that the frenemy can still cause you to recover the incorrect message $m' = m + \Delta \in \mathbb{F}$, for any $\Delta \in \mathbb{F}$ of the frenemy's choosing.

(c) You come up with an improved factoring algorithm that is too complicated to describe in a single element $m \in \mathbb{F}$. Show how you can extend Shamir secret sharing to share an element $\vec{m} = (m_1, \ldots, m_\ell) \in \mathbb{F}^\ell$ such that:

- Each friend gets a share consisting of a *single* element of $\mathbb{F}$.
- Seeing any $n - \ell$ shares leaks nothing about $\vec{m}$ (so you are protected against coalitions of $n - \ell$ frenemies who want to steal your algorithm).
- All n friends can recover the entire secret message $\vec{m}$.

You may assume that $n + 2\ell < |\mathbb{F}|$.

(d) **Extra Credit [5 points].** So far we have assumed that the number of friends n is much smaller than the size of the field $\mathbb{F}$ (i.e., we have been working modulo a prime $p \gg n$). Construct a t-out-of-n secret-sharing scheme that works when $|\mathbb{F}| < n$ such that the shares have size $\text{poly}(|\mathbb{F}|, \log n)$. If it's helpful, you may assume that $n \ll |\mathbb{F}|^2$ and have a scheme that works only for some subset of the values $t \in [n]$. There are many possible solutions to this problem, so feel free to be creative.

Problem 4: Basic Properties of SIS and LWE [15 points]. In this problem, we are going to explore the basic properties of the SIS, ISIS, and LWE problem. For the formal definitions of these problems, refer to the notes for Lecture 9 and Lecture 10. We also include them for reference at the end of this problem set.

(a) Show that solving $\mathsf{ISIS}(n, m, q, B)$ is as hard as solving $\mathsf{SIS}(n, m+1, q, B)$. In particular, show that if there is an efficient $\mathsf{ISIS}(n, m, q, B)$ adversary $\mathcal{A}$, then there exists an efficient $\mathsf{SIS}(n, m+1, q, B)$ adversary $\mathcal{B}$ such that

$$\mathsf{ISISAdv}_{n,m,q,B}[\mathcal{A}] \leq \mathsf{SISAdv}_{n,m+1,q,B}[\mathcal{B}].$$

(b) Consider the following *matrix* variant of LWE:

MatrixLWE(n, m, q, χ_B): Let $n, m, q, B \in \mathbb{N}$ be positive integers, and let χ_B be a B-bounded distribution over $\mathbb{Z}_q$. For a given adversary $\mathcal{A}$, define the following two experiments.

Experiment b $\quad (b = 0, 1)$:

- The challenger computes

$$\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{m \times n}, \quad \mathbf{S} \xleftarrow{R} \mathbb{Z}_q^{n \times n}, \quad \mathbf{E} \leftarrow \chi_B^{m \times n}, \quad \mathbf{B}_0 \leftarrow \mathbf{A} \cdot \mathbf{S} + \mathbf{E}, \quad \mathbf{B}_1 \xleftarrow{R} \mathbb{Z}_q^{m \times n},$$

  and gives the tuple $(\mathbf{A}, \mathbf{B}_b)$ to the adversary.

- The adversary outputs a bit $\hat{b} \in \{0, 1\}$.

Let W_b is the event that $\mathcal{A}$ outputs 1 in Experiment b. Then, we define $\mathcal{A}$'s advantage in solving the matrix-LWE problem for the set of parameters n, m, q, χ_B to be

$$\mathsf{MatrixLWEAdv}_{n,m,q,\chi_B}[\mathcal{A}] := \Big|\Pr[W_0] - \Pr[W_1]\Big|.$$

Show that MatrixLWE(n, m, q, χ_B) is as hard as solving LWE(n, m, q, χ_B). In particular, show that if there is an efficient MatrixLWE(n, m, q, χ_B) adversary $\mathcal{A}$, then there exists an efficient LWE(n, m, q, χ_B) adversary $\mathcal{B}$ such that

$$\mathsf{MatrixLWEAdv}_{n,m,q,\chi_B}[\mathcal{A}] \leq n \cdot \mathsf{LWEAdv}_{n,m,q,\chi_B}[\mathcal{B}].$$

[**Hint**: Use a hybrid argument.]

(c) Assume that $2mB^2 < q$ and q is prime. Show that SIS(n, m, q, B) is as hard as solving MatrixLWE(n, m, q, χ_B). In particular, show that if there is an efficient SIS(n, m, q, B) adversary $\mathcal{A}$, then there exists an efficient MatrixLWE(n, m, q, χ_B) adversary $\mathcal{B}$ such that

$$\mathsf{SISAdv}_{n,m,q,B}[\mathcal{A}] \leq \mathsf{MatrixLWEAdv}_{n,m,q,\chi_B}[\mathcal{B}] + \mathrm{negl}(n).$$

[**Hint**: Given a MatrixLWE(n, m, q, χ_B) challenge $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times n}$, start by using $\mathcal{A}$ to find a short vector $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = \mathbf{0} \pmod{q}$.]

(d) **Extra Credit [5 points].** Assume that $m = 2n \lceil \log q \rceil$. Show that solving ISIS$(n, m, q, B)$ is as hard as solving SIS$(n, m, q, 2B)$. In particular, show that if there is an efficient ISIS(n, m, q, B) adversary $\mathcal{A}$, then there exists an efficient SIS$(n, m, q, 2B)$ adversary $\mathcal{B}$ such that

$$\mathsf{ISISAdv}_{n,m,q,B}[\mathcal{B}] \leq \mathsf{SISAdv}_{n,m,q,2B}[\mathcal{A}] + \mathrm{negl}(n).$$

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?

Computational Problems. We review the formal definitions of the SIS, ISIS, and LWE problems from lecture below:

$\mathsf{SIS}(n, m, q, B)$: Let $n, m, q, B \in \mathbb{N}$ be positive integers. For a given adversary $\mathcal{A}$, we define the following experiment:

- The challenger samples $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, and gives $\mathbf{A}$ to the adversary $\mathcal{A}$.
- The adversary $\mathcal{A}$ outputs some *non-zero* vector $\mathbf{x} \in \mathbb{Z}_q^m$.

We define $\mathcal{A}$'s advantage in solving the SIS problem for the set of parameters n, m, q, B, denoted $\mathsf{SISAdv}_{n,m,q,B}[\mathcal{A}]$, to be the probability that $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod{q}$ and $\|\mathbf{x}\|_\infty \leq B$.

$\mathsf{ISIS}(n, m, q, B)$: Let $n, m, q, B \in \mathbb{N}$ be positive integers. For a given adversary $\mathcal{A}$, we define the following experiment:

- The challenger samples $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $\mathbf{y} \xleftarrow{R} \mathbb{Z}_q^n$, and gives $(\mathbf{A}, \mathbf{y})$ to the adversary $\mathcal{A}$.
- The adversary $\mathcal{A}$ outputs some *non-zero* vector $\mathbf{x} \in \mathbb{Z}_q^m$.

We define $\mathcal{A}$'s advantage in solving the ISIS problem for the set of parameters n, m, q, B, denoted $\mathsf{ISISAdv}_{n,m,q,B}[\mathcal{A}]$, to be the probability that $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \pmod{q}$ and $\|\mathbf{x}\|_\infty \leq B$.

$\mathsf{LWE}(n, m, q, \chi_B)$: Let $n, m, q, B \in \mathbb{N}$ be positive integers, and let χ_B be a B-bounded distribution over $\mathbb{Z}_q$. For a given adversary $\mathcal{A}$, we define the following two experiments:

Experiment b $\quad (b = 0, 1)$:

- The challenger computes

$$\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{m \times n}, \quad \mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n, \quad \mathbf{e} \leftarrow \chi_B^m, \quad \mathbf{b}_0 \leftarrow \mathbf{A} \cdot \mathbf{s} + \mathbf{e}, \quad \mathbf{b}_1 \xleftarrow{R} \mathbb{Z}_q^m,$$

  and gives the tuple $(\mathbf{A}, \mathbf{b}_b)$ to the adversary.
- The adversary outputs a bit $\hat{b} \in \{0, 1\}$.

Let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. Then, we define $\mathcal{A}$'s advantage in solving the LWE problem for the set of parameters n, m, q, χ_B to be

$$\mathsf{LWEAdv}_{n,m,q,\chi_B}[\mathcal{A}] := \Big| \Pr[W_0] - \Pr[W_1] \Big|.$$