CS 355: Topics in Cryptography **Spring 2019**

Problem Set 4

Due: May 13, 2019 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) Suppose n parties give their data to a trusted curator, who answers an analyst's query q using an ϵ-differentially private mechanism M. If the n parties do not want to trust the curator, but still retain ϵ-differential privacy, they can use a secure MPC protocol where $n+1$ parties (the data holders and the analyst) jointly compute M over the n data points and query q.

(b) Let $\mathbb{G}$ be a cyclic group of prime order q with a generator $g \in \mathbb{G}$ and $H\colon \mathbb{G} \to \{1,2,3\}$ be a random function. A walk on $\mathbb{G}$ defined as $x_0 \xleftarrow{R} \mathbb{G}$ and $x_{i+1} \leftarrow x_i \cdot g^{H(x_i)}$ collides in $O(\sqrt{q})$ steps in expectation (i.e., if $i_{\text{col}} = \min\{i \in \mathbb{N}\colon \exists j < i \text{ s.t. } x_i = x_j\}$, then $\mathbb{E}_{x_0,H}[i_{\text{col}}] \le O(\sqrt{q})$).

(c) Let E be the elliptic curve $y^2 = x^3 + 7x + 12$ and $P \in E(\mathbb{F}_{103})$ be the point whose x-coordinate is equal to 19. It holds $3P = P$.

(d) Consider the BLS signature scheme using the pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Say TRUE/FALSE for each of the following:

- If CDH is hard in $\mathbb{G}$ then the signature scheme is secure in the random-oracle model.
- If CDH is easy in $\mathbb{G}$ then the signature scheme is insecure.

(e) For all positive integers n, m, q, B, the $\mathsf{SIS}(n, m, q, B)$ problem is at least as hard as the $\mathsf{SIS}(n, m+1, q, B)$ problem.

Problem 2: Local Differential Privacy [10 points]. The differential-privacy model we saw in class, where a trusted curator aggregates all the data and then randomizes responses to queries, is also called the *central model* of differential privacy.

In the *local model* of differential privacy, the users do not want to trust the aggregator, so they each randomize their own data locally, before sending it to the aggregator. We'll look at a very simple local DP algorithm called *Randomized Response* (RR), which was proposed by Warner in 1965, four decades before differential privacy was invented! The goal of RR is to collect sensitive statistics (e.g., "how many people do drugs") while allowing each individual participant in the survey some amount of *deniability*.

Formally, each of the n users holds a private bit $b_i \in \{0,1\}$. The quantity we are interested in estimating is $a := \frac{1}{n}\sum_{i=1}^{n} b_i$. Consider the following RR mechanism, that is run independently by each user:

- Flip two unbiased coins.
- If the first coin is heads, send b_i to the aggregator.
- Otherwise, look at the second coin:
    - If heads, send 0 to the aggregator.
    - If tails, send 1 to the aggregator.

(a) Show that RR guarantees ϵ-differential privacy for $\epsilon = \ln(3)$ for each individual user's bit.

(b) Let $\hat{b}_i$ be the i-th user's randomized response. Show that the untrusted aggregator that receives all these noisy bits can compute an unbiased estimate $\hat{a}$ of a (i.e., $\mathbb{E}[\hat{a}] = a$).

(c) Show that the estimation error $\hat{a} - a$ has standard deviation $O(1/\sqrt{n})$.

(d) How much worse is this than what we can achieve in the central model? Suppose all users send their bits b_i to a trusted curator that uses the Laplace mechanism to output a noisy estimate $\hat{a}_\mathrm{c}$ of a that is $\ln(3)$-differentially private. Show that the estimation error $\hat{a}_\mathrm{c} - a$ has standard deviation $O(1/n)$.

(e) **Extra credit [3 points].** Design a general version of the RR mechanism that provides ϵ-differential privacy for each user's individual bit, for any fixed $\epsilon > 0$. Show that your mechanism satisfies ϵ-DP in the local model and that the standard deviation of the untrusted aggregator's estimation error is $O\left(\frac{1}{\epsilon\sqrt{n}}\right)$.

Problem 3: Discrete Log in $\mathbb{Z}_p^*$ [12 points]. The discrete-log algorithms we have seen so far are *generic*, in that they work for every group. There are beautiful special-purpose algorithms for solving discrete log faster in $\mathbb{Z}_p^*$ (for prime p). We sketch some of the ideas behind these non-generic discrete-log algorithms, and we hope to explain why they have these funny sub-exponential running times.

Let $p = 2q + 1$ be a prime such that q is also prime. Let $g \in \mathbb{Z}_p^*$ be a generator of the order-q subgroup of $\mathbb{Z}_p^*$.

(a) Say that an integer is B-smooth if it factors into primes less than B. A good approximation is that:

$$\Pr_{x \xleftarrow{R} \mathbb{Z}_q}\left[(g^x \bmod p) \text{ is } B\text{-smooth}\right] \approx u^{-u} \qquad \text{where} \qquad u = \frac{\ln p}{\ln B}.$$

Let $B(p) = \exp(\sqrt{\ln p \cdot \ln \ln p})$. Show that the probability that a random element of the subgroup of in $\mathbb{Z}_p^*$ generated by g is $B(p)$-smooth is at least $\Omega(1/B(p))$.

(b) You are given:

- an integer $h = g^x \in \mathbb{Z}_p^*$,
- all of the primes $(\pi_1, \dots, \pi_k)$ of size at most $B(p)$, and
- the discrete logs $(\log_g \pi_1, \dots, \log_g \pi_k)$ of these small primes modulo p (assume for simplicity that all of these discrete logs exist).

Give an algorithm that uses this information to recover $x \in \mathbb{Z}_q$ with constant probability in time $\mathrm{poly}(B(p))$. You should show that your algorithm is correct and that it runs in the stated time.

(c) **Extra credit [6 points].** Show how to generate the discrete logs needed for Part (b) in time $\mathrm{poly}(B(p))$. This shows that it's possible to compute discrete logs in $\mathbb{Z}_p^*$ in time $\mathrm{poly}(B(p)) = \exp(O(\sqrt{\log p \log \log p}))$.

Problem 4: Somewhat-homomorphic encryption from pairings [10 points]. In this problem, you will construct a "somewhat homomorphic" public-key encryption scheme: it allows computing any number of additions and a single multiplication. Let $\mathbb{G}_1$ be a cyclic group of prime order p and $g \in \mathbb{G}_1$ be a generator of the group. Consider the following two algorithms:

$\mathsf{Gen}(g) \to (\mathsf{pk}, \mathsf{sk})$: Choose random $a, b, c \xleftarrow{R} \mathbb{Z}_p$ such that $c \neq ab \pmod{p}$. Set $g_a = g^a$, $g_b = g^b$, and $g_c = g^c$. Output the public key $\mathsf{pk} = (g, g_a, g_b, g_c)$ and the secret key $\mathsf{sk} = (a, b, c)$.

$\mathsf{Enc}(\mathsf{pk} = (g, g_a, g_b, g_c), m) \to \mathsf{ct}$: Given a message $m \in \mathbb{Z}_p$, choose $r \xleftarrow{R} \mathbb{Z}_p$ and output $\mathsf{ct} = (g^m g_a^r, g_b^m g_c^r)$.

(a) Give a Dec algorithm that takes a secret key sk and a ciphertext $\mathsf{ct} = (u, v)$ and outputs m. Your algorithm needs to be efficient only if the message m lies in some known small space (say $0 \leq m < B$ as an integer, for some bound $B = O(\text{polylog}(p))$).

(b) Give an algorithm $\mathsf{Add}(\mathsf{pk}, \mathsf{ct}, \mathsf{ct}') \to \mathsf{ct}_{\text{sum}}$ that takes as input two ciphertexts ct and ct', that are encryptions of $m, m' \in \mathbb{Z}_p$ respectively, and outputs an encryption of $m + m' \bmod p$.

Now let $\mathbb{G}_2, \mathbb{G}_T$ be two other cyclic groups of order p (i.e., $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T|$), $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing, and $h \in \mathbb{G}_2$ and $e(g, h) \in \mathbb{G}_T$ be generators of $\mathbb{G}_1$ and $\mathbb{G}_T$ respectively. Furthermore, let $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{Gen}(h)$ be the public and secret keys obtained by running Gen using the group $\mathbb{G}_2$. Consider now the following algorithm:

$\mathsf{Mult}(\mathsf{ct}, \mathsf{ct}')$: On input two ciphertexts $\mathsf{ct} = (u, v) \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$ and $\mathsf{ct}' = (u', v') \leftarrow \mathsf{Enc}(\mathsf{pk}', m')$, output the tuple $(w_1, w_2, w_3, w_4) \in \mathbb{G}_T^4$ where

$$w_1 = e(u, u'), \quad w_2 = e(u, v'), \quad w_3 = e(v, u'), \quad w_4 = e(v, v').$$

(c) Let $\alpha_1, \ldots, \alpha_4 \in \mathbb{Z}_q$ such that $w_i = e(g, h)^{\alpha_i}$ (i.e., α_i is the discrete log of w_i in $\mathbb{G}_T$). Show that $m \cdot m' \bmod p$ can be expressed as a *linear function* of $\alpha_1, \ldots, \alpha_4$. (You *need not* give an explicit formula for the linear function.)

(d) Show how to efficiently recover $m \cdot m' \bmod p$ from $w_1, \ldots, w_4$ and the two secret keys sk and sk'. As in Part (a), you can assume that the messages m, m' lie in some known small space.

(e) **Extra credit [3 points].** Show that if the DDH assumption holds in $\mathbb{G}_1$ then $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a semantically secure public-key encryption scheme.

Problem 5: Short Integer Solutions [10 points]. (We have included a few useful definitions from lecture as an appendix to this problem set.)

(a) Give an efficient algorithm for $\mathsf{SIS}(n, m, q, q)$.

(b) Construct a function $f_{n,m,q}\colon \mathcal{X}_{n,m,q} \to \mathcal{Y}_{n,m,q}$ such that an adversary that breaks the one-wayness of $f_{n,m,q}$ with probability ϵ can be used to solve $\mathsf{SIS}(n, m, q, 1)$ with probability ϵ. (Think of n, m, and q as all being polynomial in the security parameter λ.) For the function $f_{n,m,q}$ you construct, you should specify the domain $\mathcal{X}_{n,m,q}$ and the codomain $\mathcal{Y}_{n,m,q}$. You *need not* formally prove the security of your construction.

(c) Let $H\colon \{0,1\}^n \to \{0,1\}^{\sqrt{n}}$ be a collision-resistant hash function (CRHF). Given an n-bit string as input, let there exist an algorithm that computes H in time $T(n)$.

We say that H is "updatable" if, for all $x, y \in \{0,1\}^n$ of Hamming distance one (i.e., that differ only in a single bit position), there is an algorithm U that maps:

$$U(x, H(x), y) \mapsto H(y),$$

and that runs in time $o(T(n))$. Notice that standard CHRFs, such as SHA2, are not updatable—if you change one bit of the input you have to recompute the output from scratch.

In lecture, we saw a very elegant SIS-based construction of a collision-resistant hash function H_{SIS}. Show that the H_{SIS} is updatable. In particular, construct the algorithm U, explain its running time, and argue that computing U is faster than recomputing H_{SIS}.

(d) **Extra credit [5 points]**. Show how to build an updatable collision-resistant hash function $H_{\text{big}} : \{0,1\}^{n^2} \to \{0,1\}^{\sqrt{n}}$ from a standard collision-resistant hash function $H : \{0,1\}^n \to \{0,1\}^{\sqrt{n}}$.

(e) **Extra credit [1 points]**. Give a one-sentence definition of an updatable digital signature scheme, following the definition of updatable CRHFs. Explain how to use your solution to Part (b) to construct an updatable digital signature scheme.

Problem 6: 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.

Problem 7: 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?

Useful definitions

$\mathsf{SIS}(n, m, q, B)$: 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 $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_q^{n\times m}$, and gives $\mathbf{A}$ to the adversary $\mathcal{A}$.
- The adversary $\mathcal{A}$ outputs some *non-zero* vector $\mathbf{x} \in \mathbb{Z}^m$.

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 $\|\mathbf{x}\|_\infty \leq B$.

Recall that, for a vector $\mathbf{x} = \langle x_1, \ldots, x_m \rangle \in \mathbb{Z}^m$, the L_∞-norm of the vector $\mathbf{x}$ is $\|\mathbf{x}\|_\infty := \max_i |x_i|$.

A function ensemble $\{f_\lambda : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ is *one way* if:

1. there exists an efficient algorithm that takes as input λ (represented in unary), an $x \in \mathcal{X}$, and outputs $f_\lambda(x)$ and
2. for every efficient algorithm $\mathcal{A}$,

$$\Pr\left[f(\mathcal{A}(f(x))) = f(x) :\ x \xleftarrow{\text{R}} \mathcal{X}_\lambda\right] \leq \text{negl}(\lambda).$$