CS 355: Topics in Cryptography **Spring 2024**

Problem Set

Due: Friday, 7 June 2024 (**at most 1 late day**) (submit via Gradescope)

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

https://crypto.stanford.edu/cs355/24sp/homework.tex

Submission Instructions: You must submit your problem set via Gradescope. Please use course code **RKN4PX** to sign up. Note that Gradescope requires that the solution to each problem starts on a **new page**.

You will be able to use a **maximum of 1 late day** on this assignment, and we will **not be granting extensions**.

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 Ed.

Problem 1: Verifiable Secret Sharing [10 points]. Consider a dealer who wants to share a secret α between n shareholders using the t-out-of-n Shamir secret-sharing scheme, for some $t < n$. The shareholders suspect that the dealer secretly holds a grudge against one of them and has given that person an invalid share, inconsistent with the rest of the shares. (We say that a set of shares is consistent if there exists a secret α such that every coalition of at least t shareholders can recover the (same) secret α.) In this problem, we assume that all shareholders are honest.

(a) Show that if they are willing to reveal all their shares, the shareholders can detect if one of them has indeed been given an invalid share.

We would like the shareholders to be able to detect an invalid share without having to reconstruct the secret in the verification process. To do this, consider the following modification to Shamir's secret-sharing scheme:

> Let $\mathbb{G}$ be a cyclic group of prime order $q > n$, and let g, h each be a generator of $\mathbb{G}$.
>
> 1. The dealer chooses $\beta, a_1, b_1, \ldots, a_{t-1}, b_{t-1} \xleftarrow{R} \mathbb{Z}_q$ and constructs the polynomials $A(x) = \alpha + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$ and $B(x) = \beta + b_1 x + b_2 x^2 + \cdots + b_{t-1} x^{t-1}$ over $\mathbb{Z}_q$.
> 2. The dealer creates t Pedersen commitments $c_0, c_1, \ldots, c_{t-1} \in \mathbb{G}$ where $c_o = \mathsf{Commit}(\alpha; \beta) = g^{\alpha} h^{\beta}$ and $c_j = \mathsf{Commit}(a_j; b_j) = g^{a_j} h^{b_j}$ for $j \in [t-1]$. The dealer publicly broadcasts all the commitments to all the shareholders.
> 3. The dealer creates n shares $\{(i, s_i, r_i)\}_{i=1}^{n}$, where $s_i = A(i)$ and $r_i = B(i)$ are computed over $\mathbb{Z}_q$. The dealer privately sends each of the n shareholders her own share.

(b) Describe a verification routine that allows the shareholders to jointly verify that all the shares given to them are valid without revealing any additional information about the secret.

(c) Prove that the protocol preserves the secrecy of the secret α against any coalition of fewer than t shareholders. [**Hint:** Specify the view of any coalition of $t-1$ shareholders and then prove this view is distributed independently of the secret α.]

(d) **Extra Credit [3 points].** Prove that if a dealer can trick the shareholders into accepting an invalid set of shares it can solve the discrete log of h with respect to g. A set T of shares is *valid* if for all size-t subsets $S, S' \subset T$, reconstruction gives the same result when run with S and with S'.

Problem 2: Private Information Retrieval [15 points]. Throughout this question, we consider one-round information-theoretic PIR over an n-bit database.

In class, we saw a simple two-server PIR with $O(n^{1/2})$ communication complexity. In this problem, you will first construct a *four*-server PIR scheme with communication complexity $O(n^{1/3})$. Then you will construct a *two*-server PIR with much improved $O(n^{1/3})$ communication complexity. As we mentioned in lecture, this $O(n^{1/3})$ scheme was essentially the best-known two-server PIR scheme for many many years, so in this problem you will reprove a very nice and very non-trivial result.

(a) In the following box, we describe a four-server PIR scheme with $O(\sqrt{n})$ communication. Prove that the scheme is correct. Explain *informally* in 2-3 sentences why the scheme is secure as long as the adversary controls at most *one* server.
(**Hint**: Using matrix notation will make your life easy. The correctness argument should not require more than a few lines of math.)

> **Four-Server $O(\sqrt{n})$-Communication PIR Scheme**
>
> Write the n-bit database as a matrix $X \in \mathbb{Z}_2^{\sqrt{n}\times\sqrt{n}}$. The client wants to read the bit X_{ij} from this database, where $i, j \in [\sqrt{n}]$. Recall that $e_i \in \mathbb{Z}_2^{\sqrt{n}}$ is the dimension-$\sqrt{n}$ vector that is zero everywhere except with a "1" at position i.
>
> - $\mathsf{Query}(i, j) \to (q_{00}, q_{01}, q_{10}, q_{11})$.
>   Sample random vectors $r_0, r_1, s_0, s_1 \in \mathbb{Z}_2^{\sqrt{n}}$ subject to $r_0 + r_1 = e_i \in \mathbb{Z}_2^{\sqrt{n}}$ and $s_0 + s_1 = e_j \in \mathbb{Z}_2^{\sqrt{n}}$.
>   For $b_0, b_1 \in \{0,1\}$, let $q_{b_0 b_1} \leftarrow (r_{b_0}, s_{b_1})$.
>   Output $(q_{00}, q_{01}, q_{10}, q_{11})$.
> - $\mathsf{Answer}(X, q) \to a$.
>   Parse the query q as a pair (r, s) with $r, s \in \mathbb{Z}_2^{\sqrt{n}\times 1}$.
>   Return as the answer the single bit $a \leftarrow r^T X s \in \mathbb{Z}_2$.
> - $\mathsf{Reconstruct}(a_{00}, a_{01}, a_{10}, a_{11}) \to X_{ij}$.
>   Output $X_{ij} \leftarrow a_{00} + a_{01} + a_{10} + a_{11} \in \mathbb{Z}_2$.

(b) Say that you have a k-server PIR scheme that requires the client to upload $U(n)$ bits to each server and download one bit from each server. Explain how to use this scheme to construct a k-server PIR scheme in which, for any $\ell \in \mathbb{N}$, each client uploads $U(n/\ell)$ bits to each server and downloads ℓ bits from each server. (You may assume that n is a multiple of ℓ.)

Sketch—without a formal proof—why your construction does not break the correctness or security of the initial PIR scheme.

(c) Show how to combine parts (a) and (b) get a four-server PIR scheme with total communication $O(n^{1/3})$. In particular, you should calculate the optimal value of the parameter ℓ used in part (b).

(d) Sketch how to generalize the PIR scheme in part (a) to give an eight-server PIR scheme in which the client sends $O(n^{1/3})$ bits to each server and receives a single bit from each server in return. This should only take a few sentences to describe.

(e) Now comes the grand finale! Use the *eight*-server scheme from part (d) to construct a *two*-server scheme with communication $O(n^{1/3})$.

Hint:

- Label the queries of the eight-server scheme from part-(d) as $q_{000}, q_{001}, q_{010}, \ldots, q_{111}$. The two queries in your new two-server scheme should be q_{000} and q_{111} from the eight-server scheme.
- The two servers can clearly send back the 1-bit answers for q_{000} and q_{111} respectively. NOW, here is the beautiful idea: show that by sending back to the client $O(n^{1/3})$ additional bits, each of the two servers can enable the client to recover the answers for three additional queries.

(f) **Extra credit [2 points].** Show how to construct a 4-server PIR scheme with $O(\sqrt{n})$ communication that is secure against any coalition of up to 3 servers. (For the correctness property to hold, all 4 servers might still need to be honest.)

Problem 3: Key-Exchange from LWE [18 points]. In this problem, we will formalize the concept of a *non-interactive key exchange* (NIKE) protocol, and then construct it from LWE. NIKE protocols are a core component of Internet protocols like TLS, and the lattice-based NIKE that we develop in this problem is a simplified variant of some of the leading candidates in the NIST competition for standardizing post-quantum key-exchange.

A *non-interactive key exchange* (NIKE) protocol for a key space $\mathcal{K}$ consists of the following PPT algorithms:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: On input the security parameter λ, the setup algorithm outputs the public parameters pp.
- $\mathsf{ClientPublish}(\mathsf{pp}) \to (\mathsf{priv}, \mathsf{pub})$: On input the public parameters pp, the client-publish algorithm outputs a secret value priv, and a public message pub.
- $\mathsf{ServerPublish}(\mathsf{pp}) \to (\mathsf{priv}, \mathsf{pub})$: On input the public parameters pp, the server-publish algorithm outputs a secret value priv, and a public message pub.
- $\mathsf{KeyGen}(\mathsf{priv}, \mathsf{pub}) \to \mathsf{key}$: On input a secret value priv, and a public message pub, the key generation algorithm outputs a key $\mathsf{key} \in \mathcal{K}$.

Correctness. We require that when $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pub}_0, \mathsf{priv}_0) \leftarrow \mathsf{ClientPublish}(\mathsf{pp})$, $(\mathsf{pub}_1, \mathsf{priv}_1) \leftarrow \mathsf{ServerPublish}(\mathsf{pp})$, we have

$$\Pr\left[\mathsf{KeyGen}(\mathsf{priv}_0, \mathsf{pub}_1) = \mathsf{KeyGen}(\mathsf{priv}_1, \mathsf{pub}_0)\right] = 1 - \mathrm{negl}(\lambda)$$

where the probability is taken over the randomness of all procedures.

Security. For a NIKE protocol $(\mathsf{Setup}, \mathsf{ClientPublish}, \mathsf{ServerPublish}, \mathsf{KeyGen})$, we define the following two experiments:

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

- The challenger computes the following:
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$,
  $(\mathsf{priv}_0, \mathsf{pub}_0) \leftarrow \mathsf{ClientPublish}(\mathsf{pp})$,
  $(\mathsf{priv}_1, \mathsf{pub}_1) \leftarrow \mathsf{ServerPublish}(\mathsf{pp})$,
  $\mathsf{key}_0 \leftarrow \mathsf{KeyGen}(\mathsf{priv}_0, \mathsf{pub}_1)$,
  $\mathsf{key}_1 \xleftarrow{\mathrm{R}} \mathcal{K}$.

  It provides $(\mathsf{pp}, \mathsf{pub}_0, \mathsf{pub}_1, \mathsf{key}_b)$ 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 say that a NIKE protocol is secure if

$$\left|\Pr[W_0] - \Pr[W_1]\right| = \mathrm{negl}(\lambda).$$

(a) Explain in words why the security definition above captures our intuitive notion of security for key-exchange.

(b) Consider the following NIKE protocol:[1]

> Let $n = \text{poly}(\lambda), q, \chi_B$ be parameters for which $\mathsf{LWE}_{\mathsf{HNF}}(n,n,q,\chi_B)$ and $\mathsf{LWE}_{\mathsf{HNF}}(n,n+1,q,\chi_B)$ is hard. Recall from lecture that in practice, for $\lambda = 128$, we use $n \approx 800$.
>
> Define the key space $\mathcal{K} = \{0,1\}$ and consider the following algorithms.
>
> - $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: Sample a matrix $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_q^{n\times n}$ and set $\mathsf{pp} = \mathbf{A}$.
> - $\mathsf{ClientPublish}(\mathsf{pp}) \to (\mathsf{priv}, \mathsf{pub})$: Sample vectors $\mathbf{s} \leftarrow \chi_B^n$, $\mathbf{e} \leftarrow \chi_B^n$. Then, set $\mathsf{priv} = \mathbf{s}$, and $\mathsf{pub} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$.
> - $\mathsf{ServerPublish}(\mathsf{pp}) \to (\mathsf{priv}, \mathsf{pub})$: Sample vectors $\mathbf{s} \leftarrow \chi_B^n$, $\mathbf{e} \leftarrow \chi_B^n$. Then, set $\mathsf{priv} = \mathbf{s}$, and $\mathsf{pub} = \mathbf{A}\mathbf{s} + \mathbf{e}$.
> - $\mathsf{KeyGen}(\mathsf{priv}, \mathsf{pub}) \to \mathsf{key}$: Let $\mathsf{priv} = \mathbf{s} \in \mathbb{Z}_q^n$ and $\mathsf{pub} = \mathbf{b} \in \mathbb{Z}_q^n$. The key generation algorithm first samples a small noise term $e \leftarrow \chi_B$. Then, if $\|\langle \mathbf{s}, \mathbf{b}\rangle + e\|_\infty \le \lfloor q/4 \rceil$, set $\mathsf{key} = 0$. Otherwise, set $\mathsf{key} = 1$.

Here, $\lfloor q/4 \rceil$ denotes the integer closest to $q/4$, with ties broken downward. Suppose that q is prime and chosen to satisfy $4nB^2/q = \text{negl}(\lambda)$. Prove that the protocol satisfies correctness. For the proof, feel free to use the following fact (you do not need to prove this fact):

> For any prime q, for $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_q^{n\times n}$ any two non-zero vectors $\mathbf{s}_0, \mathbf{s}_1 \in \mathbb{Z}_q^n$, and $c \in \mathbb{Z}_q$,
>
> $$\Pr_{\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}}\left[\mathbf{s}_0^T \mathbf{A} \mathbf{s}_1 = c\right] = 1/q - \text{negl}(\lambda),$$
>
> where the probability is over the random choice of $\mathbf{A}$.

(c) Prove that the protocol above is secure assuming $\mathsf{LWE}_{\mathsf{HNF}}(n,n,q,\chi_B)$ and $\mathsf{LWE}_{\mathsf{HNF}}(n,n+1,q,\chi_B)$. The definition of $\mathsf{LWE}_{\mathsf{HNF}}$ is on the last page of this problem set. [**Hint:** Use a hybrid argument.]

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?

[1] We restrict the key space to $\mathcal{K} = \{0,1\}$ for simplicity. To get a NIKE protocol for $\mathcal{K} = \{0,1\}^{128}$, we can simply run 128 parallel instances of the protocol using the same public matrix $\mathbf{A}$.

(d) Do you have any other feedback on the course so far?

Problem 4: Time Spent [1 point 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.

Appendix: Definition of LWE in Hermite Normal Form.

We review the formal definitions of the Learning with Errors problem in *Hermite Normal Form*. Note that in this variant of the LWE problem, the vector $\mathbf{s}$ is sampled from the B-bounded error distribution χ_B instead of the uniform distribution. This version of the LWE problem is known to be as hard as the standard LWE problem.

$\mathsf{LWE}_{\mathsf{HNF}}(n, m, q, \chi_B)$: 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

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

  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 $\mathsf{LWE}_{\mathsf{HNF}}$ problem for the set of parameters n, m, q, χ_B to be

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