Homework 4

Due date: Tuesday, April 25, 2023 at 11:59PM (Gradescope)

Question 1 (30 points)

Consider the Single-Server Private-Information Retrieval problem where Bob (server) has a database $D = \{x_1, \dots, x_n\}$. Alice would like to retrieve the item x_i without revealing i to Bob. Formally, a solution consists of three PPT algorithms ($\texttt{Query}, \texttt{Respond}, \texttt{Recover}$). Here, $\texttt{Query}$ takes as input a security parameter λ (unary), an index $i \in [n]$, and random coins R and outputs a query q and a hint s to be used later i.e., $(q, s) = \texttt{Query}(1^\lambda, i; R)$. $\texttt{Respond}$ takes as input a query q and the database $D = \{x_1, \dots, x_n\}$ and generates a response $r = \texttt{Respond}(q, D)$. Finally, $\texttt{Recover}(r, s)$ takes as input a response r and a hint s and outputs a value x.

Usage Intuitively, Alice is given i and generates $(q, s) = \texttt{Query}(1^\lambda, i; R)$. Alice sends the query q to Bob who will respond with $r = \texttt{Respond}(q, D)$. Finally, Alice recovers $x_i = \texttt{Recover}(r, s)$.

Correctness The scheme is correct if for any database D of n items $x_1, \dots, x_n$, any security parameter λ and any random coins R and any index $i \in [n]$ we have $\texttt{Recover}(r, s) = x_i$ where $(q, s) = \texttt{Query}(1^\lambda, i; R)$ and $r = \texttt{Respond}(q, D)$.

Security The scheme is secure if for all PPT distinguishers $\mathcal{A}$ there is a negligible function $\mu(\dots)$ such that for all λ and all indices $i, j \in [n]$ we have

$$\left| \Pr_R \left[\mathcal{A}\left(1^\lambda, q\right) = 1 \ : (q, s) \leftarrow \texttt{Query}(i; R) \right] - \Pr_R \left[\mathcal{A}\left(1^\lambda, q\right) = 1 \ : (q, s) \leftarrow \texttt{Query}(j; R) \right] \right| \leq \mu(\lambda) .$$

Part A. Consider the Pallier construction described informally in the slides. Prove that this scheme is correct and secure and analyze the computational/communication overhead for both parties. The construction is described more formally below.

$\texttt{Query}(1^\lambda, i; (R_1, R_2))$ works as follows 1) Generate a Pallier Key $(pk, sk) = \texttt{PKeyGen}(1^\lambda, R_1)$ using random coins R_1, 2) Set $c_i = \texttt{Enc}_{pk}(1)$ and $c_j = \texttt{Enc}_{pk}(0)$ for $j \neq i$, 3) Set $q = (pk, c_1, \dots, c_n)$ and $s = sk$ and return (q, s).

$\texttt{Respond}(q, x_1, \dots, x_n)$ works as follows 1) parse q to extract $(pk, c_1, \dots, c_n)$ and extract N from the Pallier key pk, 2) compute $c'_j = c_j^{x_j} \mod N^2$ (Note: you may assume that $x_j < N$ for each $j \in [n]$), 3) Compute $r = \prod_{j=1}^{n} c'_j \mod N^2$ and return r.

$\texttt{Recover}(r, s) \doteq \texttt{Dec}_s(r)$.

> *Answer:*
>
> ...

Part B. Assume that we have Fully Homomorphic Encryption (FHE). Develop a secure PIR protocol which reduces the communication and computation overhead for Alice.

> *Answer:*
>
> ...

Part c. Prove your construction in part B is secure.

> *Answer:*
>
> ...

> *Resource and Collaborator Statement:*
>
> ...

Question 2 (40 points)

Consider a quantum attacker $\mathcal{A}$ who has quantum access to the random oracle $H : \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ and classical access to the oracle $H(K,\cdot)$ where $K \in \{0,1\}^\lambda$ is a uniformly random key (unknown to the attacker $\mathcal{A}$). In many applications it makes sense to assume that $\mathcal{A}$ only has classical access to the latter oracle $H(K,\cdot)$ e.g., because the attacker can only observe the response $H(K,x)$ if it convinces the honest party to encrypt a classical message related to x. We define two hybrids: H_0 and H_1. In H_0 (real world) we pick K randomly the attacker gets quantum access to $H(\cdot)$ and classical access to the oracle $H(K,\cdot)$ as above. In H_1 the attacker still gets quantum access to $H(\cdot)$, but the oracle $H(K,\cdot)$ is replaced by a truly random function $f : \{0,1\}^\lambda \to \{0,1\}^\lambda$ which is unrelated to $H(\cdot)$. Let p_0 (resp. p_1) denote the probability that $\mathcal{A}$ outputs 1 in hybrid H_0 (resp. H_0) then the advantage of the attacker is $\mathtt{ADV}_{\mathcal{A}} = |p_0 - p_1|$.

Part A (10 points) As a warm-up suppose that $\mathcal{A}^{H(\cdot)}$ makes at most T queries to $H(\cdot)$ and only has access to the oracle $H(\cdot)$ i.e. $\mathcal{A}$ makes no queries to $H(K,\cdot)$. Let $\psi_0^s, \psi_1^s, \ldots, \psi_T^s$ denote the states after each query to the random oracle $H(\cdot)$ when we run $\mathcal{A}^{H(\cdot)}(s)$ on initial input s. For each key $K' \in \{0,1\}^\lambda$ let $S_{K'} = \{(K',x) : x \in \{0,1\}^\lambda\}$. Given a quantum state $\phi = \sum_{x,y,z} \alpha_{x,y,z} |x,y,z\rangle$ let $\mathtt{QM}(K',\phi) \doteq \sum_{x,y,z:x\in S_{K'}} |\alpha_{x,y,z}|^2$ denote the magnitude on basis states where we are making a query of the form $H(K',\cdot)$. We say that a key K' is ϵ-bad for the pair $(s, H(\cdot))$ if

$$\sum_{i=0}^{T-1} \mathtt{QM}(K', \psi_i^s) \geq \epsilon \ .$$

Formally, let $\mathtt{K}_{\epsilon,s,H} = \left\{K' : \sum_{i=0}^{T-1} \mathtt{QM}(K',\psi_i^s) \geq \epsilon\right\}$ denote the set of ϵ-bad keys K'. Fix any pair (s,H) and upper bound $|\mathtt{K}_{\epsilon,s,H}|$ the number of ϵ-bad keys. Your upper bound should be a function of T and ϵ.

Answer: ...

Part B. (10 points) Let $F : \{0,1\}^\lambda \to \{0,1\}^\lambda$ be any function and let $H_{F,K}(\cdot)$ denote an oracle such that $H_{F,K}(K,x) = F(x)$ and $H_{F,K}(K',x) = H(K',x)$ whenever $K' \neq K$. Let $s_F \in \{0,1\}^{\lambda 2^\lambda}$ be a bit string describing the truth table of $F(\cdot)$.

Suppose that $K \notin \mathtt{K}_{\epsilon,s_F,H}$. Upper bound the Euclidean distance between $\psi_t^{s_F}$ (the final state when we run $\mathcal{A}^{H(\cdot)}(s_F)$) and $\psi_{t,K}^{s_F}$ (the final state when we run $\mathcal{A}^{H_{F,K}(\cdot)}(s_F)$)

Answer: ...

Part C. (10 points) Assume that the attacker $\mathcal{A}$ makes at most q_1 quantum queries to $H(\cdot)$ and at most q_2 classical queries to the second oracle (either $H(K,\cdot)$ or $f(\cdot)$). Upper bound $\mathtt{ADV}_{\mathcal{A}}$.

Answer:

...

Part D. (10 points) Consider the encryption scheme $\texttt{Enc}_K(m) = (r, H(K, r) \oplus m)$. Argue that the scheme is CPA-Secure in the Quantum Random Oracle Model.

Answer:

...

Resource and Collaborator Statement:

...

Question 3 (30 points)

In this problem we consider the Private Two-Server Keyword Search problem. Suppose that two servers B and C each hold a copy of the database $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_1, \ldots, x_n$ denote distinct keywords and $y_1, \ldots, y_n \in \{0,1\}^m \setminus \{0^m\}$ denote documents. Alice A would like to search for a specific keyword x and retrieve the associated document y if the pair $(x, y) \in D$ appears in the database. Alice does not want server B or C to learn the value of the query x. This rules out a naive protocol where Alice send x to either server. However, Alice does trust that servers B and C will not communicate.

Part A. Formalize the intuitive security property i.e., provide a formal security definition (Concrete/Asymptotic style definitions are both acceptable)

> *Answer:*
>
> ...

Part B. Define a secure two-server protocol using Distributed Point Functions. For full credit you should make sure that Alice's computational/communication complexity remains as low as possible. **Note:** You may assume that the Distributed Point Function shares f_1 and f_2 of the point function $f_{\alpha,\beta}(\cdot)$ [1] which, on input x, output additive shares $f_1(x)$ and $f_2(x)$ such that $f_1(x) + f_2(x) = f_{\alpha,\beta}(x) \mod 2^m$ for all inputs x.

> *Answer:*
>
> ...

Part C. Argue that your protocol is secure.

> *Answer:*
>
> ...

> *Resource and Collaborator Statement:*
>
> ...

[1] Recall that the point function $f_{\alpha,\beta}(\cdot)$ is defined as follows $f_{\alpha,\beta}(\alpha) = \beta$ and $f_{\alpha,\beta}(x) = 0$ for all inputs $x \neq \alpha$.