Privacy Enhancing Technologies FS2025
Exercise Sheet 5 (version 1)

Florian Tramèr

Problem 1: Conceptual Questions. For each of the following statements, say whether it is TRUE or FALSE. Write at most one sentence to justify your answer.

(a) With recursive PIR, we can get a 2-server information-theoretic PIR scheme with $O(\text{polylog} n)$ communication complexity.

(b) Assume the binary database X stored in the single-server PIR scheme from the lecture is *sparse*, i.e., $X_i = 1$ for only $o(n)$ indices i. Then the server's work can be sublinear in n.

(c) In a secure ORAM for a memory of n words, for every read operation, the RAM must perform $O(n)$ operations to avoid leaking information about the memory access.

(d) A secure ORAM protocol implies a secure single-server PIR protocol.

Problem 2: PIR from Distributed Point Functions Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite fields. For $x \in \mathcal{X}, y \in \mathcal{Y}$, the point function $P_{x,y} : \mathcal{X} \to \mathcal{Y}$ is defined by $P_{x,y}(x) = y$ and $P_{x,y}(x') = 0$ for all $x' \neq x$.

A *distributed point function* (DPF) is a succinct additive secret-sharing of a point function, i.e., a way to create keyed functions $\texttt{Eval}(k_0, \cdot)$ and $\texttt{Eval}(k_1, \cdot)$ such that $\texttt{Eval}(k_0, \cdot) + \texttt{Eval}(k_1, \cdot) = P_{x,y}(\cdot)$.

Definition 1 (Distributed Point Function)**.** A distributed point function (for two servers) is a pair of poly-time algorithms $(\texttt{Gen}, \texttt{Eval})$ with the following syntax:

- $\texttt{Gen}(x, y)$ where $x \in \mathcal{X}, y \in \mathcal{Y}$ outputs a pair of keys (k_0, k_1).
- $\texttt{Eval}(k, x')$ where $k \in \{0,1\}^*, x' \in \mathcal{X}$ outputs $y' \in \mathcal{Y}$.

The DPF is secure if it satisfies the following property:

- **Correctness**: For all $x, x' \in \mathcal{X}$, $y \in \mathcal{Y}$, and $(k_0, k_1) \leftarrow \texttt{Gen}(x, y)$:

$$\texttt{Eval}(k_0, x') + \texttt{Eval}(k_1, x') = \begin{cases} y & \text{if } x' = x \\ 0 & \text{otherwise} \end{cases}$$

- **Secrecy**: For $\beta \in \{0,1\}$ there exists a simulator $\texttt{Sim}_\beta$ such that for all $x \in \mathcal{X}, y \in \mathcal{Y}$:

$$\texttt{Sim}_\beta(|x|, |y|) \overset{c}{\approx} \{k_\beta \ : \ (k_0, k_1) \leftarrow \texttt{Gen}(x, y)\} \ .$$

(a) Given a DPF with $\mathcal{X} = \mathbb{F}_{2^{\log n}}, \mathcal{Y} = \mathbb{F}_2$ (i.e., strings of $\log n$ bits and 1 bit respectively, with addition done modulo 2) and keys of size ℓ bits, construct a computationally secure two-server PIR protocol for a database of size n with $O(\ell)$ communication complexity.

(note that the most efficient DPFs today have keys of size $\ell = O(\lambda \log |\mathcal{X}|)$ bits, and this yields the most efficient computational 2-server PIR schemes to date.)

(b) In the *PIR with keywords* setting, both servers hold a dictionary of the form (w_i, v_i) for $i \in [n]$, where the keywords $w_i \in \{0,1\}^\omega$ are all of length ω bits, and the values $v_i \in \{0,1\}^v$ are of length v bits. Given some keyword w', the client wants to obtain the corresponding value (or 0^v if w' is not in the dictionary).

Describe a 2-server PIR protocol for this setting using an appropriate DPF. You can assume the key size of your DPF is ℓ bits. Your scheme should have communication complexity $O(\ell + v)$.

(c) Does the 2-server scheme we described in the lecture allow you to implement a PIR by keyword scheme? Why or why not?

Problem 3: A 2-Server Information-theoretic PIR with $O(n^{1/3})$ Communication. 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.

- 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 $(q_{00}, q_{01}, q_{10}, q_{11})$.
- 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 $a \leftarrow r^T X s \in \mathbb{Z}_2$.
- 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 $O(n^{1/3})$.

Hints:

- Label the queries of the eight-server scheme from part-(d) as $q_{000}, q_{001}, q_{010}, \dots, q_{111}$. 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.

Problem 4: Maliciously secure ORAM For this problem, you can assume we use the $\sqrt{n}$ ORAM from the lecture, although the problem applies to any ORAM. Suppose the data in physical RAM is encrypted with a semantically secure encryption scheme with key k, where k is stored in the ORAM client.

The problem is that this ORAM provides no integrity protection: the adversary (i.e., the RAM server) can respond to a `Read` query with any value it wants.

(a) As a partial solution, suppose we add a MAC to the data.[1] That is, when the ORAM client wants to write value `data` to address a, it first computes $m = \text{MAC}(k, (a, \texttt{data}))$ and asks the server to store $(\texttt{data}, m)$ at address a. When the client reads from address a, it asks the server to return the pair $(\texttt{data}, m)$ and then checks that $m = \text{MAC}(k, (a, \texttt{data}))$. If not, the client aborts. Show that this scheme is insecure: there exist programs where the server can respond to a `Read` query with an incorrect value that the client will accept.

(b) Propose a protocol that is maliciously secure, in that the client never accepts an incorrect value from the RAM. You can assume that when performing a $\texttt{Read}(a)$ or $\texttt{Write}(a, \texttt{data})$ operation, the ORAM client can easily check how many previous reads and writes it has done for address a during the execution of the program.

[1] A MAC is a keyed function $m \leftarrow \text{MAC}(k, \texttt{data})$ such that it is hard for an adversary to compute a correct MAC value for a given message without knowing the key k (they are thus essentially a symmetric-key variant of digital signatures).