# CS 65500
# Advanced Cryptography

## Lecture 23: Private Information Retrieval

Instructor: Aarushi Goel

Spring 2025

# Agenda

→ Definition, Motivation

→ K-server PIR

→ Single server PIR from additively homomorphic encryption.

→ Damgård - Jurik encryption.

## Homework 6:

Q1 We can assume $H$ is a random oracle, i.e, returns random outputs. Therefore $Pr[H(x) = H(y)] \leq \frac{1}{|\mathbb{F}|}$

Q2 Remember in PCG, we want the output of Setup to be sublinear in the length of vector $\vec{a}, \vec{c}$.

Q3 Observe that in this question, you are effectively showing that linearly homomorphic secret-key encryption (with some additional special properties) is equivalent to public-key encryption.
Such equivalence does not hold for regular secret-key encryption schemes. There are known separation results.

# Private Information Retrieval (PIR)

Server

Client

| | Server | | Client |
|---|---|---|---|
| Input: | $\boxed{d_1 \mid \ldots \mid \ldots \mid \cdots \mid d_N}$ | | $i \in [N]$ |
| | database | $\longleftarrow$ $\vdots$ | index |
| Output: | $\perp$ | $\longrightarrow$ | $d_i$ |

* Correctness: client learns the desired record $d_i$.

* Security: the (malicious) server should learn nothing about $i$.

We do not require privacy for server's DB. Otherwise, this would be equivalent to OT.

## Trivial Solution

→ Since we do not care about privacy for the server, a trivial approach would be to let the client download the entire DB.

→ Server's communication : $O(N)$

* Goal: The goal is to minimize the size of server's response to the client. Hence we want to design more efficient constructions.

## Applications.

If we can do this, we can use PIR as the basic building block for several privacy-preserving protocols, with applications in :

* private DNS lookup
* safe browsing
* private contact tracing
* contact discovery
* anonymous messaging.

Q: Can we design PIR schemes where the computation time for the server is sublinear in $N$?

A: No! It has to be atleast linear.

If it were sublinear, that would mean some records in the DB we ignored and the server will learn they are NOT $d_i$.

Recent Breakthrough: Doubly-efficient PIR. (2022)

Server can do some preprocessing on the DB. Subsequently all queries can be answered in sublinear time.

By Wei-Kai Lin, Ethan Mook, Daniel Wichs.

(NOT TODAY)

## K-Server PIR

→ This is a relaxed version of single-server PIR, where K-servers hold copies of the same DB. The client wants to retrieve an element from this dabase

→ Security: Unless all servers collude, none of them learn any information about $i$.

Q: Can we build 2-server PIR using any of the primitives that we have discussed in this course so far?

A: Yes, using 2-party distributed point functions.

How? (Think!!)

# Single-Server PIR using Additively Homomorphic Encryption.

→ Let's assume all elements in the database $\in \mathbb{Z}_p$.

→ Let (Gen, Enc, Dec) be an additively homomorphic public-key encryption scheme with message space $\mathbb{Z}_p$.

Server

Client

Input: $d_1, \ldots, d_N$      $i$

$\forall j \in [N], j \neq i \quad ct_j = Enc(pK, 0)$

$ct_i = Enc(pK, 1)$

$\xleftarrow{\quad ct_1, \ldots, ct_N \quad}$

$ct = \sum_{j \in [N]} d_j \cdot ct_j$

$\xrightarrow{\quad ct \quad}$

$Dec(sK, ct) \rightarrow d_i$

Problem: Server communication is sublinear, but client's communication is larger than the DB.

# Single-Server PIR with Sublinear Client Communication (Candidate?)

Server

Client

Input: $d_1, \ldots, d_N$ (Server) $\qquad i$ (Client)

$$\begin{array}{c} \\ i^* \\ \\ \end{array} \boxed{\begin{array}{ccc} d_1, \ldots & & d_{\sqrt{N}} \\ \vdots & & \\ & \boxed{d_i} & \\ \vdots & & \\ \vdots & \ldots & d_N \end{array}}$$

$i^* = \lfloor i/\sqrt{N} \rfloor$

$\forall j \in [\sqrt{N}]\ j \neq i^* \quad ct_j = Enc(pk, 0)$

$ct_{i^*} = Enc(pk, 1)$

$j^* = i \bmod \sqrt{N}$

$\forall k \in [\sqrt{N}],\ k \neq j^* \quad \bar{ct}_k = Enc(pk, 0)$

$\bar{ct}_{j^*} = Enc(pk, 1)$

$$\xleftarrow[\bar{ct}_1, \ldots, \bar{ct}_{\sqrt{N}}]{ct_1, \ldots, ct_{\sqrt{N}}}$$

$$\forall j \in [\sqrt{N}]: \quad A_j = \Big(\sum_{k \in [\sqrt{N}]} d_{j+k}\Big) \cdot ct_j$$

$$A = \sum_{j \in [\sqrt{N}]} \bar{ct}_j \times A_j$$

$$\xrightarrow{\quad A \quad}$$

$Dec(sk, Dec(sk, A)) \rightarrow d_j$

We can also recurse on this idea.

# Final PIR scheme

→ We can recursively use the idea discussed earlier as follows:

$ct = \mathrm{Enc}(\mathrm{Enc}(\mathrm{Enc}(d_5))) \longrightarrow \mathrm{Dec}(\mathrm{Dec}(\mathrm{Dec}(ct))) = d_5$

$\mathrm{Enc}(\mathrm{Enc}(d_1)) \quad \mathrm{Enc}(\mathrm{Enc}(d_5)) \longleftarrow \mathrm{Enc}(0),\ \mathrm{Enc}(1)$

$\mathrm{Enc}(d_1) \quad \mathrm{Enc}(d_3) \quad \mathrm{Enc}(d_5) \quad \mathrm{Enc}(d_7) \longleftarrow \mathrm{Enc}(1),\ \mathrm{Enc}(0)$

$d_1 \quad d_2 \quad d_3 \quad d_4 \quad d_5 \quad d_6 \quad d_7 \quad d_8 \longleftarrow \mathrm{Enc}(1),\ \mathrm{Enc}(0)$

Server

Client

Lets assume the client's input $i=5$

→ problem with this approach is that each $A_j$ is itself a ciphertext. As a result, $A_j$ might not be in $\mathbb{Z}_p$.

→ Unless $A_j$ can be efficiently mapped to an element in $\mathbb{Z}_p$, we cannot rely on the homomorphic properties of the encryption scheme that has message space $\mathbb{Z}_p$ to compute $A = \sum \bar{c}t_j \cdot A_j$

What we want: a *recursive* homomorphic encryption scheme where ciphertext in one level is plaintext in the next level.
To recursively use of this idea, we additionally want the ciphertext size to only increase *additively* from level to level.

## Damgård-Jurik Encryption Scheme.

→ Based on the *decisional composite residuosity* assumption (DCR)

→ Additively homomorphic.

→ Can be used to encrypt messages $\in \mathbb{Z}_{n^s}$.

→ elements in $\mathbb{Z}_{n^s}$ can be represented using $s.\log n$ bits.

→ $s \log n$ bits are encrypted to a ciphertext of size $(s+1)\log n$ bits

→ Generalization of Paillier's encryption scheme.