Lecture 3: Private Information Retrieval
Plan

* Recap: Preliminaries
* PIR: What it is, why it's amazing
* Stretch break
* Constructions
  - Two-server PIR
  - One-server PIR

Logistics

* HW1 due this Friday 9/18 @ 5pm Boston via Grockscape
  ➔ You must use Latex template
* OHI: W 3-4:30pm on Zoom (link on Piazza)
* Please give feedback on psets
* Anonymous Feedback Form
Recap: Fundamental Primitives

PRPs $\xrightarrow{\text{switching lemma}}$ PRFs $\xrightarrow{\text{counter mode}}$ PRGs $\xrightarrow{\text{immediate}}$ OWFs

- $\text{PRPs} \xrightarrow{\text{Luby-Rackoff}} \text{Feistel} \xrightarrow{\text{GGM tree}} \text{OWFs}$

$\rightarrow$ All imply each other.

$\rightarrow$ None imply key exchange.

- If $P = NP$, none exist.
- If $P \neq NP$?

Asymptotic view:
- Security parameter $\lambda$
- Efficient $= \text{poly}(\lambda)$
- Small $= \text{reg}(\lambda)$

Concrete view:
- Efficient = runs on your computer in reasonable time bound
- Small $= 2^{-128}$
A "perfect" research result

1) Has a beautiful theory
2) Works in practice
3) Solves a problem that people care about.

⇒ It's a rare piece of work that meets this rubric. But aim high.

Today

- One of my favorite "almost perfect" ideas in crypto.
- Lots of activity, more ongoing even today.
  ➡ Will cover recent results next week.
- A classic crypto result: seems impossible, then simple.

Bad news: For reasons we'll see, it's not quite practical yet.
Private Information Retrieval

Every day on the Internet

Client

query

response

Server
Examples

1. DNS

```
mit.edu
```

```
184.87.51.152
```

2. "Fever"

```
<article about fever>
```

```
WebMD
```

3. Many more...

- Searching Google
- Looking news articles
- Fetching data from social media networks
- Looking for Airbnb properties

Nota: The client's query can be sensitive!
It can leak:

- what website you're visiting
- your health conditions
- your travel plans
- political interests
- ...

Today, servers learn all of these things!
**Question:** "Can you query a database without the database learning your query?"

**Trivial answer:** "Yes, just download the entire database."

DB server doesn’t learn your query.

Still, this is unsatisfying.

Let’s ask a better question..."
Better Question:

"Can you query a database without the database learning your query... with communication sublinear in the database size?"

Answer:

Unconditionally, no. \([\text{CGKS '95}]\)

We won't prove this, but there is a clean information theoretic argument in the original PIR paper.

What do we do when we are stuck in life?

Option I: Change the model. \((\text{Ron, etc.})\)

What if we have two non-colluding copies of the DB?

"two-server PIR" \([\text{CGKS '95}]\)

Option II: Make assumptions!

Under basic "public-key" assumptions \((\text{DDH, Paillier...})\) we can build non-trivial PIR on 1 server

"single-server PIR" \([\text{Kushilevitz & Ostrovsky '97}]\)

\((k\text{-server PIR uses } k \text{ DB replicas})\)
Important! Security only holds if servers do not collude (i.e., one of the two servers is honest).

Non-essential simplifications

* DB is an array of bits (can extend to handle longer rows)

* DB lookup is by index (can implement a key-value map)
More formally:

Two-server PIR consists of three eff algs:

\[ \text{Query}(1^n, i) \rightarrow (q_0, q_1) \]
\[ \text{Answer}(x, q_0) \rightarrow a \]
\[ \text{Reconstruct}(a_0, a_1) \rightarrow x_i \]

Properties

1. Correctness: Client gets the bit it wants.
\[ \forall n \in \mathbb{N}, \forall i \in \mathbb{N}, \forall x \in \{0, 1\}^n \]
\[ P_r \left[ \text{Reconstruct}(a_0, q_0) = x_i : \begin{aligned} (q_0, q_1) &\leftarrow \text{Query}(1^n, i) \\ a_0 &\leftarrow \text{Answer}(x, q_0) \\ a_1 &\leftarrow \text{Answer}(x, q_1) \end{aligned} \right] = 1 \]

2. No single server learns anything about client bit.
\[ \forall n \in \mathbb{N} \forall i, j : i \neq j \in [n], \forall a \in \{0, 1\} \]
\[ \{q_0 : (q_0, q_1) \leftarrow \text{Query}(1^n, i) \} \approx_c \{q_0 : (q_0, q_1) \leftarrow \text{Query}(1^n, j) \} \]

Non-collusion is captured by our requirement that the marginal distributions are indist.

\[ \Rightarrow \text{In info-theoretic setting (explain), } (q_0, q_1) \]
\[ \text{will leak secret index } i. \]
Stretch Break
Two-server PIR scheme with $O(n^2)$ communication.

Idea: View database as a matrix in $\mathbb{F}_2^{m \times n}$.

Client wants to read bit $(i,j) \in [\lceil \sqrt{n} \rceil] \times [\lceil \sqrt{n} \rceil]$.

Query $(1^n, i,j) \rightarrow (q_0, q_1)$

Sample random $q_0, q_1 \in \mathbb{F}_2^n$ s.t.

$q_0 + q_1 = e_i \in \mathbb{F}_2^n$

Answer $(x, q) \rightarrow x \cdot q \in \mathbb{F}_2^n$

Reconstruct $(a_0, a_1) \rightarrow (a_0 + a_1)_i = x_i$
1. Correctness.

\[ a_0 + a_1 = (Xq_0 + Xq_1); \]
\[ = (X(q_0 + q_1)); \]
\[ = (Xe_i); \]
\[ = x_i \]

2. Security

\[ q_0 \text{ is a uniform random vector (independent of } (i,j)). \]
\[ \Rightarrow \text{Same for } q_1. \]

Notice: No computation assumptions here!

Efficiency: Upload: \( 2\sqrt{n} \) bits
Download: \( 4\sqrt{n} \) bits.
\( \Rightarrow \text{Total: } 4\sqrt{n} \) bits.
**Single-server PIR**

Linearly homomorphic encryption scheme:

\[ E(k, m_0) + E(k, m_1) = E(k, m_0 + m_1) \mod 2. \]

Can build from QR, DDH, LWE, ...

Idea: Client sends encryption of its query vector rather than using secret sharing.

\[ E(k, c_j) = (E(k, 0), \ldots, E(k, 1), E(k, 0), \ldots) \]

Output \[ x_j \leftarrow (\text{Dec}(k, a)) \]

\[ = \text{Dec}(k, \text{Enc}(k, x_j)) \]

\[ = x_j \]

Communication: \( 2\sqrt{n} \) ciphertexts.
Reducing the communication

Let's look more closely at our PIR scheme...

Let's consider the following equation:

\[ X = D \left( \begin{array}{c} u \\ \vdots \\ u \text{ ctests} \end{array} \right) \]

Client throws away all but one of the responses!

Idea: View answer to query as another database and run a second PIR on this DB.
Communication fell to \( \approx n^{1/3} \).

Then recurse?
The only catch is that each step of the recursion

\[ n \text{ bits} \rightarrow \sqrt{n} \text{ ciphertexts}. \]

Under "reasonable" assumptions (or takes \(2^{\tilde{O}(n^3)}\) time),

\[ \gtrsim 2^{O(\sqrt{\log n \log \log n})} \]

gain communication.

With more esoteric cryptosystems (based on Phi-hiding, Damgård–Jurik), can drop comm cost to \(\text{polylog}(n)\).
State of the art in PIR

* Two-server PIR

Information theoretic - $O(\sqrt{\ell \log \ell / \log n}) = n^{o(1)}$

Do better schemes exist? With $O(\log^2 n)$ comm.?

Computational - $O(1 \log n)$

Boyle speaking here next week!

* Requires only PRGs. Concretely quite efficient.

* Single-server PIR

poly log(n) communication - from QR, ODF, LWE,.....

(Carribel, Miccoli, Staller '19, Lipmaa '05, ...
Computational Efficiency in PIR

In all above schemes, the servers run in time linear in the DB size.

⇒ Linear DB scan per query.

For certain “natural” PIR schemes, this limitation is inherent. (Bein et al., Ishi, Malkir ‘04)

⇒ We will see some ways to reduce the computation cost at the servers.