Lesson 20 Lecture Notes

CS555

1 Private Information Retrieval (PIR)

1.1 Definition

A client wishes to retrieve $D[i]$ from a server holding database D while hiding the index i.

- **Correctness:** Client obtains the desired bit.
- **Security:** Server cannot distinguish between queries for $D[i]$ and $D[j]$.
  - Queries must include some randomization factor.

1.2 Applications

- Private metadata messaging: sender places message in memory location where recipient looks (via PIR), instead of using unencrypted headers.
- Private search engines.

1.3 Defining a PIR Solution

- **Trivial Solution:** Client downloads the entire database.
- **Goal Operation:** fast server computation and small (sublinear) computation
- **Attaining Security:**
  - **Info-theoretic security:** only possible with (a) full communication (entire database) OR (b) by enforcing non-collusion on two servers
  - **Computational Security:**
    - can work without database replication and without non-colluding assumption
    - based in computational assumptions
    - expensive server cost because of cryptographic operations

1.4 Two-Server Non-Colluding Construction

- Client maps query index i to one-hot vector v_i.
- Randomly split v_i into two shares: $v_i = q_1 + q_2$.
  - alone they both look random; together they give v_i
- Query DB_1 with q_1, DB_2 with q_2.
- Each server computes $\langle x, q_j \rangle$.
- Client combines results: $\langle x, q_1 \rangle + \langle x, q_2 \rangle = \langle x, v_i \rangle = x_i$.
- This is still n bits of communication → no better than the full download

1.5 Matrix Form

- Database represented as $\sqrt{(n)}$x$\sqrt{(n)}$ matrix M.
- Client downloads column containing desired entry.
- Secret shares: $v_i = q_1 + q_2$.
- DB_1 returns Mq_1, DB_2 returns Mq_2.
- Client computes $Mq_1 + Mq_2 = Mv_i$, yielding the ith column.
- Can manipulate DB size for tradeoff between communication and computation size

1.6 Alternative Construction

- Client has set S of size $\approx \frac{n}{2}$.
- If $i \in S$, define $S_1 = S - \{i\}$, $S_2 = S$.

- DB_1 gives $\sum_{j \in S_1} x_j$, DB_2 gives $\sum_{j \in S_2} x_j$.
- Client computes difference to obtain x_i.
- if $i \notin S \rightarrow$ add in i and then do the same thing
- Computation $\rightarrow \frac{n}{2 \log n}$
- less computation: have more servers with overlapping queries
- 2^d servers $\rightarrow O(n^{1/4})$ communication

1.7 Multi-Server Information-Theoretic PIR

- Two servers achieve $O(n^{1/3})$ communication.
- Generally use secret sharing (additive or Shamir) to split index into two.

1.8 Single-Server PIR

- Based on computational hardness assumptions.
- Typically expensive due to cryptographic operations.
- Constructions use homomorphic encryption:
  - Linear HE: $\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$ AND $a \cdot \text{Enc}(x) = \text{Enc}(ax)$
- **Homomorphic Construction:**
  - Client encodes one-hot query vector v_i with encryption $(e_i = \text{Enc}(v_i))$
  - Server computes encrypted inner product with database $(\langle x, e_i \rangle)$
  - Client decrypts to obtain x_i.
  - here you are sending n bits and receiving 1 back
  - Matrix form: $M \cdot q$ returns an encrypted column which can then be accessed further
  - Requires HE scheme supporting multiplication.

1.9 Regev's Homomorphic Encryption

- Bulk on the computations can be preprocessed
- Public random matrix A.
- Encryption: $\text{Enc}(s, v) = (A, As + e + v)$.
- Used to encrypt full vector v_i to get $e_i = \text{Enc}(v_i)$
- Have $De_i = (DA, D(As + e + v_i))$
- Most computation (DA) can be preprocessed.