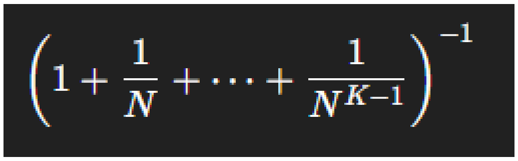
Theoretic- Privacy Information Retrieval

1. Introduction:

A cryptographic technique that allows the client to retrieve information from the server, without revealing to the server which information was revealed. It was introduced by Chor et. al. in 1995 in FOCS, in a landmark paper called Privacy Information Retrieval. A story-theory for this idea goes like: "You want the librarian to give you a specific book, but you don't want the librarian to know which book." Later in 2016, Sun and Jafer published a landmark paper in theoretic PIR, deriving the formula for Optimal Rate in PIR.

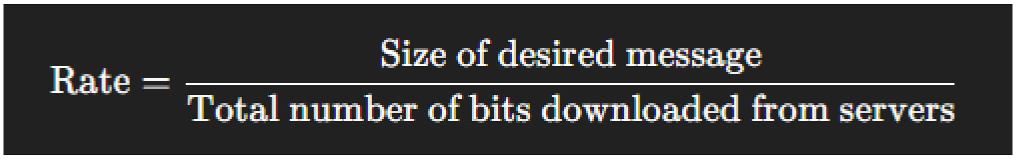
2. Theoretic PIR:

- Chor et al. in 1996 in FOCS presented the first protocol for Information-Theoretic PIR using multiple non-colluding servers and formalized the PIR model. They showed that, just one server requires multiple theoretical assumptions, ,which led to development of computational PIR.
- Later in 2016, Hua Sun and Syed A. Jafar in their paper, The Capacity of PIR formalized the formula , where N is number of servers and K is the number of queries.

  $$\left(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-1}}\right)^{-1}$$

- The trivial formula for this rate can also be given by:

  $$\text{Rate} = \frac{\text{Size of desired message}}{\text{Total number of bits downloaded from servers}}$$

3. Trivial Example for Capacity of PIR:

3.1 Objects in the space:

- There exists 2 Non-Colluding Duplicate Servers, with 2 files name W_2 and W_1.
- There exists a client, that queries to the server for retrieval of information.

3.2 Assumptions:

- The files W_2 and W_1. are divided into 4 sub-parts namely W_{11}, W_{12} , W_{13} W_{14} and W_{21}, W_{22} , W_{23} W_{24}. Each of these parts can be thought of first 25 pages of a book, then the next 25 pages of the book and so on and so forth.
- Servers are non-colluding and not byzantine.

3.3 PIR Steps

- In the first Retrieval: W_{11}, W_{21} are retrieved from the first server and W_{12}, W_{22} are retrieved from the second server. No privacy is loss, as an equal amount of W_2 and W_1 are retrieved and the server is "honest but curious"
- In the second Retrieval, W_{13} and W_{22} are retrieved from the first server and W_{14} and W_{21} are retrieved from the second server.

3.4 Rate:

- Hence retrieving W_1 completely, while still maintaining privacy and having a rate of 2/3, since 4 correct files were retrieved, and 6 files were downloaded.
- This can be verified using the Optimal Rate formula as well.

4. Theoretic PIR: Probabilistic Model:

4.1 Core Idea:

- When referring to probabilistic PIR, we're referring to PIR schemes that involve randomness either in query generation or servers response.
- A Probabilistic PIR uses a randomized query in form of a one hot encoded vector for the number of bits in the server. Since, the query is randomized it looks statistically the same to servers, without revealing any information.

4.2 Objects and Assumption:

- Lets assume, there are 2 non-colluding duplicate byzantine servers.
- The servers contain 7 bits (D_1, D_2... D_7)
- A client who sends a randomized vector for the 7 bits.

4.3 Choosing a Random Vector:

Index (j)	Q1[j]	Q2[j]
0	0	0
1	1	1
2	0	1
3	0	0
4	1	1
5	0	0
6	1	1

- In the 2 queries only the third bit(2nd indexed) is different. This is the file we want to query.

4.4 Server Computation:

- Let’s assume a sample database:: `D = [1, 0, 1, 1, 0, 1, 1]`
- Server Q1 receives: `Q1 = `0100101`
- Dot product of D and Q1:
  $D.Q1 = (0{\times}1) + (1{\times}0) + (0{\times}1) + (0{\times}1) + (1{\times}0) + (0{\times}1) + (1{\times}1) = 0 + 0 + 0 + 0 + 0 + 0 + 1 =$
  > 1
- Let this product be $a1 = 1$
- Similar Dot product of D and Q2:
  $D.Q2 = (0{\times}1) + (1{\times}0) + (1{\times}1) + (0{\times}1) + (1{\times}0) + (0{\times}1) + (1{\times}1) = 0 + 0 + 1 + 0 + 0 + 0 + 1 =>$
- Let this product be $a2 = 2$
- Now we compute the mod of a1 and a2 by 2:
  $a1 \mod 2 => 1$
  $a2 \mod 2 => 0$

4.5 Decoding by Client:

- d_2 = 1 by doing $a_1 \oplus a_2 = 1 \oplus 0 = 1$

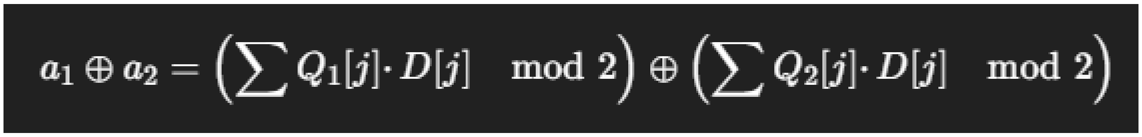
4.6 Key Idea

The PIR trick is to make the queries Q_1 and Q_2 identical at every index except for the one you want — let's call it index i (say i = 2).

That means:

- For all $j \neq i$, $Q_1[j] = Q_2[j]$
- For $j = i$, $Q_1[i] \neq Q_2[i]$ (one has a 1, the other has a 0)

This design ensures that the dot products a_1 and a_2 include contributions from all bits in the database (where the query has 1), but when you XOR a_1 and a_2:



$$a_1 \oplus a_2 = \left(\sum Q_1[j] \cdot D[j] \mod 2\right) \oplus \left(\sum Q_2[j] \cdot D[j] \mod 2\right)$$

All the matching bits (i.e. where $Q_1[j] = Q_2[j]$) will cancel out when XORed.
Only the contribution from d_i (the one bit where Q_1 and Q_2 differ) will survive in the final XOR.

5. Areas of Exploration:

1. **Colluding Servers:** When the servers collude, which is a realistic aspect, it becomes harder to maintain privacy and an optimal rate.
2. **Byzantine Servers:** When the server lies, and is not honest
3. **Weak Privacy:** - some information leakage is allowed — the server may learn a little about what the user is retrieving, Full privacy is expensive; weak privacy trades off a small amount of leakage for improved efficiency (rate).

6. Applications of PIR:

1. **Private Search Engines**: Retrieve search results **without revealing your query** to the search engine. Ex) : You want to search "symptoms of depression" privately.
2. **Encrypted Cloud Storage Access:** Access your own data from a cloud provider **without revealing which file** you’re accessing. The cloud sees only encrypted queries.
3. **Blockchain and Web3:** Query a smart contract or blockchain database **without revealing which address or token** you're interested in. Used in **private DeFi**, voting, or NFT marketplaces.
4. **Healthcare and Genomics:** Researchers can query genomic databases or medical records **without exposing which gene/disease** they are studying.
5. **Intelligence and Surveillance:** Agencies may query large datasets (e.g., leaked data, communications logs) **without exposing their target** to the data provider.
6. **Anonymous Credentials / Authentication:** Prove that you have a certain attribute (e.g., age > 18, citizen of X) **without revealing which one** you queried or which database record you matched.

7. **E-commerce Recommendations:** Ask for product suggestions **without revealing your interests** to the recommender system.
8. **Keyword Search over Encrypted Data:** Find documents matching a keyword from an encrypted dataset **without revealing the keyword** or the matched document.
