Private Information Retrieval

Shashank Hegde
Associate Professorship of Embedded Systems and Internet of Things
Technical University of Munich
Munich, Germany
shashank.hegde@tum.de

***Abstract*—Private Information Retrieval (PIR) is a cryptographic protocol that allows a client to retrieve information from a database without revealing the specific query or the data being requested to the database server. PIR has gained significant attention due to its ability to preserve privacy and confidentiality in data retrieval scenarios. However, as the scale of databases and the demand for privacy increase, new challenges arise. This paper provides a comprehensive overview of various PIR techniques, it's strengths and challenges, and an examination of the trade-offs between privacy, communication complexity, and computational overhead in designing efficient PIR schemes.**

Keywords: PIR, communication complexity, computation overhead, confidentiality

I. Introduction

The proliferation of online services and the increasing concern for data privacy have brought forth the need for efficient and secure data retrieval mechanisms. Traditional data retrieval approaches often require the disclosure of query information to the database server, which raises privacy concerns. One easy but inefficient way of ensuring data privacy and information security is by downloading the entire database of n bits and retrieving the required information, but this approach clearly has too much communication overhead. So, we need a more efficient data retrieval technique, and this is the motivation behind PIR, to enable users to access information from databases while maintaining their privacy with efficient methods. PIR offers a valuable solution for protecting sensitive user data in various applications, such as healthcare, financial systems, and cloud computing, where privacy is paramount. PIR also allows clients to retrieve data from potentially untrusted or adversarial servers without compromising privacy. This is particularly valuable in scenarios where clients have to rely on external servers or cloud platforms to store and retrieve their data, ensuring that confidentiality is maintained even in the presence of potentially malicious entities. The earliest references for PIR was in 1995 by Chor, Kushilevitz, Goldreich and Sudan [1], [2]. Through an analysis of existing PIR protocols, this paper aims to contribute to the understanding and advancement of PIR in the field of secure data management.

II. BACKGROUND

Consider a situation where the user wants file x_i from the database x. In this scenario, the identity of i must be hidden from the server. We thus require information-theoretic security. The user could either download all the files, or implement PIR techniques. Consider a situation where the user requires a bit with index $i \in [n] \triangleq \{1, \ldots, n\}$ from the database x. As i must be hidden from the serve, the user generates a random vector $r \in F^n$ over a finite field F, produces k queries (of length l_q each), q_1(i, r), . . . , q_k(i, r), one per server. The servers respond with replies A_1, . . . , A_k that depend on the contents of the database, denoted x, and the corresponding query. The user reconstructs the desired bit x_i from these k replies, together with i and r. This basic PIR approach is defined as CKGS scheme. The components of a PIR system include n servers, z colluding servers, and (n-k) unresponsive servers, also called stragglers. Several modes of PIR exist, with single server and multi-server approaches.

A. Multi-Server Approach

Consider a general PIR with stragglers, as per Fig. 1. Let any k out of n servers respond to the query of the user. Assume z<k servers are colluding, thus t=k-z, where t is the length of messages derivable. Initially, the user encodes $e_i^{(1)}, \ldots, e_i^{(t)}$ into the queries $q_1, \ldots, q_n$ using an (n,k,z) McEliese-Sarwate Secret Sharing Scheme[3] or something similar. The user sends the queries to the servers, and each server computes and sends q_jx. The user computes $x_i = (x_i^1, \ldots, x_i^t) = (e_i^{(1)}x, \ldots, e_i^{(t)}x)$ from any k results.Thus, the user downloads k symbols to retrieve the t=k-z parts of the file.

The information theoretic security would be $n^{o(1)}$ communication and computation security would be O($\log n$) communication.

B. Two Server Approach

Considering an example from the previous multi-server case, let the database be an array of n bits as a matrix, with the column and row having $\sqrt{n}$ bits each, as shown in Fig. 2. Let $x \in (0,1)^n$ as matrix $X \in Z_2^{\sqrt{n}*\sqrt{n}}$. The client wants to read $(i, j) \in [\sqrt{n}] \times [\sqrt{n}]$. The user samples two random query vectors of length $\sqrt{n}$, where the XOR of the query vectors q_1 and q_2 will be the e_j unit vector, with 0 in all positions except position j, which is one of the indices that the user wants to read. The user then sends the query vectors to the two servers, and each server takes a matrix vector product of the database and the query vector to generate another $\sqrt{n}$ answer vector which is sent to the user. During reconstruction, the user XOR's the two answer vectors and reads the i^{th} element of the resulting vector as the single bit output. The efficiency in this case is $|q_1|$ + $|q_2|$ + $|a_1|$ + $|a_2|$ = $4\sqrt{n}$
The query vectors can further be compressed to O($\log n$) using

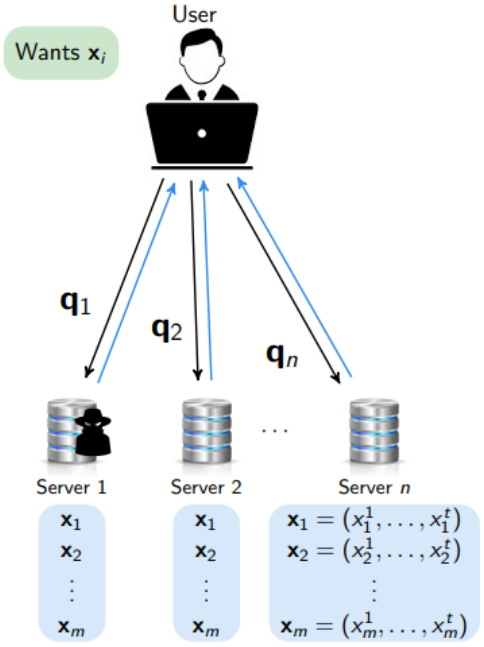

Fig. 1. General Multi-Server PIR with Stragglers

Distributed Point Functions (DPF) [4], but the system would then be based on computational security as the DPF requires Pseudo Random Generators and computational assumptions.

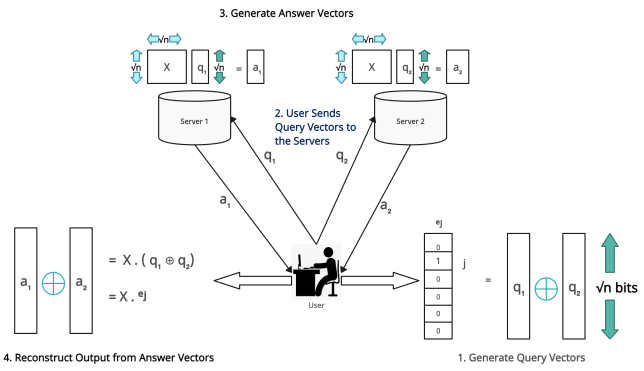

Fig. 2. Two Server Approach.

The cost of this download is half, as the user downloads two symbols to obtain one. The rate R of a PIR scheme is the number of desired symbols to the number of downloaded symbols, and the capacity is the largest achievable rate. The capacity of PIR over replicated database with n servers, n-k stragglers and z colluding servers is $C_m = \frac{1-\frac{z}{k}}{1-\left(\frac{z}{k}\right)^m}$

The capacity strictly reduces with the length of the query vector m, and when the length of the query vectors approach infinity, the capacity approaches $\frac{k-z}{k}$.

C. Single Server Approach

In a single server approach, instead of secretly sharing the vector as in the multi-server case, the query is encrypted using linearly homomorphic encryption techniques, which allow the computation of the addition of ciphertext, that encrypts the sum of the underlying plaintext.
Enc(k,m_1) + Enc(k,m_2) = Enc(k, m_1 + m_2).
The security of this approach depends on the security level of the linearly homomorphic encryption used.

III. PIR DEFINITIONS

The PIR functions should satisfy the following definitions:

Correctness: For every $x \in F^n$, $i \in [n]$, and $r \in F^n$, $Reconstruction(i, r, A_1(x, Q_1(i, r))) = x_i$

Privacy: The k-server, n-dimensional PIR scheme is defined as t-private if no collusion of up to t servers can formally learn any information about i, for any $i, j \in [n]$, with $s \in [k]$, and $q \in \{F^n\}$, the distributions of generated queries are identical
$Pr(Q_s(i, r) = q) = Pr(Q_s(j, r) = q)$
The privacy requirement is that each individual query is distributed independently of i and thus the server gains no information about the identity of the desired item (in Shannon's sense).

Communication Cost of PIR: The communication cost of a PIR scheme Π_0 over a field F is defined as:
comm(Π_0) = up(Π_0) + down(Π_0) $\triangleq$ $\max_i \sum_{j\in[k]} |q_j|$ + $max_i \sum_{j\in[k]} |a_j|$
$|q_j|$ denotes the upload cost and $|a_j|$ denotes the download cost

IV. CHALLENGES

While the previous general scheme of PIR offered privacy against a reliable server, it did not provide protection for the user/client against a malicious server. In the presence of a malicious server, the client may receive incorrect data item $\hat{x}_i \neq x_i$. To address this issue and create a secure PIR scheme that can handle malicious servers, two approaches are commonly employed: the joint-design approach, where a PIR scheme is designed with inherent security, and the modular approach, which involves combining a PIR scheme with another independently designed cryptographic primitive. Another main challenge facing PIR techniques is the high computational cost, especially on the server side. Traditional PIR schemes require linear-time server computation, which makes them impractical for large databases. Additionally, some PIR schemes require high client-side storage or communication costs, which can also be a barrier to practical implementation.

V. LITERATURE SURVEY

In the paper by Quang Cao[5] et. al., a modular approach with a commitment scheme on top of the PIR is applied, where a digest of the data which is referred to as a commitment, is published before the PIR is started. Once the commitment

has been produced and made public, the client can use the commitment to confirm the correctness of its desired data item, even when all servers are malicious.

One potential solution for verifying the correctness of derived data is through the use of cryptographic hashes as commitments. Publicly sharing the hashes h_j = h(x_j), where j ranges from 1 to n, before the PIR session starts allows the client to download all the hashes and perform a hash verification on the derived $\hat{x}_i$. If h($\hat{x}_i$) equals h_i, the client accepts it as valid. However, this solution comes with increased download costs for the client due to the additional sn hashes received from s servers, where s is a constant. Additionally, this approach makes the PIR protocol cumbersome and unsuitable for systems that require compact data commitments, such as blockchains.

To address the limitations associated with using cryptographic hashes for verification, an alternative approach using linear map commitments (LMC) [6] in conjunction with linear PIR schemes is proposed. The LMC technique enables the prover to generate a commitment C(x) for a vector $x = (x_1, x_2, ..., x_n) \in F^n$, accompanied by a witness w_a(x). This witness allows the verifier to validate if a retrieved value $y \in F$ truly corresponds to the linear combination $a \cdot x = \sum_{j=1}^{n} a_j x_j$. Employing a compact variant of LMC, which ensures that the sizes of the commitment C(x) and the witness w_a(x) (in bits) remain constant, equivalent to only a few field elements. This design choice makes the PIR scheme well-suited for databases containing large-sized data items. In this approach, $x_i \in F^m$, where m can be arbitrarily large is considered, and the LMC construction is applied to the hashes h = $(h_1, h_2, ..., h_n)$ of $x \in F^{mn}$.

In Figure 3, the client privately requests both x_1 and its corresponding hash $h_1 = h(x_1)$, utilizing the verifiability of the retrieved hash $\hat{h}_1$ through the LMC. Subsequently, the client can verify the correctness of the retrieved data $\hat{x}_1$ by comparing its hash verification $h(\hat{x}_1) \stackrel{?}{=} h_1$. The size of the x_j's can be arbitrarily large. By employing the Lai-Malavolta LMC [6], the commitment C(h) and witnesses $w_1(a_1$(h)), $w_2(a_2$(h)) can be kept compact at only 384 bits, which is only 1.5 times of SHA3-256 hash.

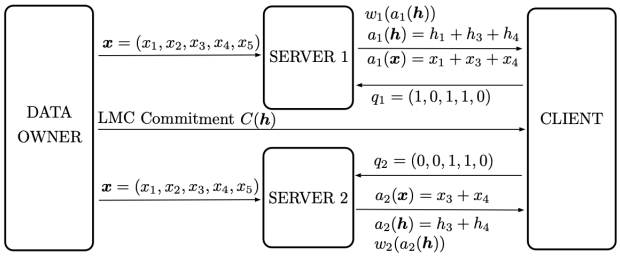

Fig. 3. Two server committed PIR scheme based on an LMC and the CKGS PIR scheme [3]

In the paper by Daniel Demmler et al [7], RAID PIR is discussed, which is a multi-server private information retrieval (PIR) scheme that enhances privacy in cloud computing. It is similar to a RAID system, where data is distributed across multiple servers for better performance. Each server stores only a part of the database, and the data from the different servers is combined in a simple, efficient operation. Some advantages of this approach include:

- Improved efficiency over known PIR protocols.
- Uses only very efficient cryptographic primitives.
- Well suited for cloud deployment as it reduces the communication as well as the computational workload per server.

The PIANO technique, described in the paper [8] is a novel single-server PIR scheme that aims to achieve sublinear server computation and optimal client storage. It is designed to enable efficient access to private information in large databases while preserving privacy. This scheme does not require any form of homomorphic encryption or other heavy-weight cryptographic primitives such as privately puncturable PRFs, and only requires pseudorandom functions (PRFs), which can be accelerated through the AES-NI instruction sets available in most modern processors. The approach is self-contained without the need to invoke any existing PIR scheme as a building block. With the sublinear server computation, PIANO can scale to handle much larger database sizes compared to previous implementations. This makes it suitable for organizations dealing with massive amounts of data. It was observed that Piano achieved a response time of 99 ms for a 100 GB database over a coast-to-coast link, with only a 1.62× slowdown compared to a non-private baseline. This demonstrates the scalability of the PIANO technique in terms of response time and performance. PIANO is designed to address the scalability challenges of PIR and offers a practical solution for accessing private information in large databases.

VI. Computation of PIR

The server linearly scans the entire database to respond to a query in a CKGS PIR, which would put huge computations on the server side. The logic is such that if the server doesn't touch the bit 'i' in the database, the user would not be reading the bit 'i', and this holds true for many non-colluding servers irrespective of the cryptographic implementation. Some of the steps to reduce the complexity at the server side include the following techniques:

A. Batch PIR

In this approach, the server and client agree on a random partition of the database into Q buckets. The client queries each bucket $\lambda \log n$ times, where λ is the security parameter that is used to control the level of privacy and security in the protocol, and n is the database size. Instead of requesting items one by one, the user submits a batch query containing multiple item requests, one for each bucket simultaneously. The server which sees $\lambda \log n$ queries for each bucket regardless of the user's input, is unaware of the specific items being requested, and returns the corresponding data to the user, preserving the

privacy of individual queries.
Batch PIR can benefit from parallel processing and optimization techniques, allowing servers to retrieve multiple items efficiently and potentially reducing the overall response time. The server time of this scheme is $\mathrm{Q} \cdot \lambda \log n \cdot \mathrm{T}\left(\frac{n}{q}\right) = \mathrm{Q} \cdot \lambda \log n \cdot \frac{n}{q} = \mathrm{n} \cdot \lambda \log n$

B. PIR Preprocessing

In this scheme, the servers pre-compute the responses to all possible queries of a standard PIR scheme. By investing computational resources in the preprocessing phase, the overall efficiency of the PIR protocol can be improved, enabling faster response times and reduced communication overhead during information retrieval. However one of the major issues is storage overhead, where the preprocessed information needs to be stored by the servers. So even though the problem of sublinear server time is solved, there is a superlinear server storage issue.

C. PIR Extensions

Realistically, data is organized in blocks [9] rather than as single bits, and the client retrieves an entire block of data from the database in a single PIR query, which reduces the communication complexity between the client and the database server.
PIR by keyword [10], another extension of Private Information Retrieval, allows the client to search for a specific string or keyword in a database. By utilizing block PIR and a binary search approach with lexicographically ordered strings, the client can retrieve only the relevant subset of data related to the queried keyword. This significantly reduces the computational complexity at the client side from O(N) to O(M), where M represents the size of the reduced subset.

VII. Conclusion

In this paper, we have explored the topic of Private Information Retrieval (PIR) and its various approaches and techniques. We have discussed the importance of PIR in preserving privacy while enabling efficient retrieval of data from remote servers. Several key conclusions can be drawn through analysis:

1. PIR schemes aim to balance efficiency and security. Advancing PIR techniques often involves finding ways to improve efficiency without compromising the security guarantees. While achieving sublinear server computation in PIANO is a significant advancement, further improvements may require careful consideration of the trade-off between efficient PIR schemes and security.

2. PIR protocols based on cryptographic primitives, such as pseudorandom functions (PRFs) and homomorphic encryption, offer effective techniques for achieving privacy without relying on heavyweight cryptographic tools.

3. Advances in PIR techniques, such as straggler mitigation and replication strategies, address challenges related to server performance and reliability, enhancing the overall efficiency and robustness of PIR systems.

4. The design and analysis of PIR protocols require careful consideration of factors like communication complexity, computational overhead, collusion resistance, and security assumptions.

5. PIR continues to be an active area of research, with ongoing efforts to improve efficiency, scalability, and practical deployment in real-world scenarios.

In conclusion, PIR provides a powerful tool for preserving privacy in information retrieval scenarios. By employing various techniques and optimizing different parameters, researchers and practitioners can continue to enhance the privacy guarantees, efficiency, and practicality of PIR systems, thereby facilitating secure and confidential data access.

References

[1] Chor, O. Goldreich, E. Kushilevitz and M. Sudan, "Private information retrieval", Proceedings of the 36th Annual Symposium on Foundations of Computer Science, pp. 41-50, 1995

[2] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan, "Private Information Retrieval", Journal of the ACM (JACM), vol. 45, no. 6, pp. 965-981, 1998

[3] R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," Commun. ACM, vol. 24, no. 9, p. 583–584, Sep 1981. https://doi.org/10.1145/358746.358762

[4] N. Gilboa, Y. Ishai, " Distributed point functions and their applications", EUROCRYPT 2014. LNCS, vol. 8441, pp. 640–658. Springer, Heidelberg 2014

[5] Quang Cao, Hong Yen Tran, Son Hoang Dau, Xun Yi, Emanuele Viterbo, Chen Feng, Yu-Chih Huang, Jingge Zhu, Stanislav Kruglik, and Han Mao Kiah, "Committed Private Information Retrieval", arXiv:2302.01733v1, Feb. 2023

[6] Lai R. and Malavolta G., "Subvector Commitments with Application to Succinct Arguments", Annual International Cryptology Conference. pp. 530–560 2019

[7] Daniel Demmler, Amir Herzberg and Thomas Schneider, "RAID-PIR: Practical multi-server PIR", Conference: Workshop on Privacy in the Electronic Society (WPES)

[8] Mingxun Zhou, Andrew Park, Elaine Shi and Wenting Zheng, "Piano: Extremely Simple, Single-Server PIR with Sublinear Server Computation", Cryptology ePrint Archive, Paper 2023/452

[9] L. Wang, T. K. Kuppusamy, Y. Liu and J. Cappos, "A Fast Multi-Server, Multi-Block Private Information Retrieval Protocol," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 2015, pp. 1-6

[10] Benny Chor, Niv Gilboa, Moni Naor, "Private Information Retrieval by Keywords", Technical report, TR CS0917, Department of Computer Science, Technion 1997

[11] Bringer J., Chabanne H., "Another Look at Extended Private Information Retrieval Protocols", Progress in Cryptology – AFRICACRYPT 2009, Volume 5580.

[12] W. Gasarch, "A Survey on Private Information Retrieval," in Bulletin of the EATCS, vol. 82, Feb. 2004, pp. 72–107.

[13] Alexandra Henzinger, Matthew M. Hong, Henry Corrigan-Gibbs, Sarah Meiklejohn, Vinod Vaikuntanathan, "One Server for the Price of Two: Simple and Fast Single Server Private Information Retrieval", Cryptology ePrint Archive, Paper 2022/949

[14] A. Beimel, Y. Ishai, and T. Malkin. "Reducing the servers' computation in private information retrieval: PIR with preprocessing", Advances in Cryptology– CRYPTO 2000, volume 1880 of Lecture Notes in Computer Science, pages 56–74. Springer, 2000

[15] H. Sun and S. A. Jafar, "The Capacity of Private Information Retrieval," IEEE Global Communications Conference (GLOBECOM), Washington DC, USA, 2016

[16] M. A. Attia, D. Kumar and R. Tandon, "The Capacity of Private Information Retrieval From Uncoded Storage Constrained Databases," IEEE Transactions on Information Theory, vol. 66, no. 11, pp. 6617-6634, 2020