MAC0499 - Proposal and Schedule
Folding FrodoPIR (FFPIR)

Luiza Barros Reis Soezima [1]

[1]University of São Paulo (USP), Instituto de Matemática e Estatística (IME)
NUSP:11221842
lbrsoezima@usp.br

1 Supervisors

- **Supervisor:** Hilder Vitor Lima Pereira [2]

  [2] Universidade Estadual de Campinas (UNICAMP), Instituto de Computação, Campinas, Brazil

  hilder@unicamp.br

- **Co-supervisor:** Alfredo Goldman [3]

  [3] University of São Paulo (USP), Instituto de Matemática e Estatística (IME)

  gold@ime.usp.br

2 Introduction

Retrieving information from databases pulls a constant activity on the daily routine, concurrently, its privacy concern when it comes to sensitive data develop a parallel problem. Private information retrieval (PIR) is a privacy protocol that allows a user to download a required message from a set of messages stored in a database without revealing the index of the required message to the databases.

In other words, PIR is protocol in which from one side, a possibly untrusted server holds a public database DB with N records. On the other side, a client wants to query for record $i \in \{0 \cdots N-1\}$, without letting the server learn the queried item they are looking up (and, hence, learning the value v associated with i they are interested in). A naive solution involves the client locally downloading the whole DB, but that can be expensive: the goal of PIR is to both preserve privacy and be more efficient than the total cost of downloading the whole DB. There are many proposed solutions for this problem, and for this Capstone Project, we will explore the ones that uses Fully Homomorphic Encryption(FHE) as cryptographic primitive. [2, 5, 9, 8, 3, 4, 11, 6, 7, 10, 1]

3 Preliminaries and Fundamentals

3.1 Stateful Private Information Retrieval

It is important to note that the PIR interaction is divided into two parts: (1) offline query-independent and (2) online query-dependent.

- **Offline Phase:**
  - `ssetup(`1^λ`)`: An algorithm that runs on the server, generating the initial parameter `ip`.
  - `cinit(ip)`: An algorithm where the client initializes with the initial parameters `ip`. It generates a message *msg* that is sent to a server during the offline phase. This phase can be omitted, making the scheme client-independent for preprocessing.
  - `spreproc(ip, DB,`*msg*`)`: Server preprocessing algorithm that runs using the initial parameters `ip`, the server's database DB, and the client's message *msg*. It generates a set of public parameters `pp` that are downloaded by the client.
  - `cpreproc(ip,pp)`: Client preprocessing algorithm that runs on the server generated with the public parameters (`ip`,`pp`) and generates a state `st`.
- **Online Phase:**
  - `query(st,`*i*`)`: Algorithm in which the client generates a query `q` for the item at index *i* in the server's DB, and optionally returns an updated state `st'`.
  - `respond(DB, q)`: Algorithm in which the server generates a response `r` to be sent to the client.
  - `process(st, r)`: Algorithm in which the client uses this response `r` and generates an element `x` from the DB.

4 FrodoPIR Original Scheme

The protocol consists of 5 parts, where:

- **Offline**
  1. In the **offline** phase, the server interprets the database as a matrix and applies a **compression** function to reduce its size, creating a global parameter. This compression function reduces the size of the DB by m/λ, where λ is the security parameter and m is the number of elements in the DB. Therefore, note that the parameter **is not** linear in the size of the DB.
  2. Also in the **offline** phase, the client downloads the public parameters and computes c sets of preprocessed query parameters.
- **Online**
  1. In the **online** phase, the client uses a set of preprocessed query parameters to create the encrypted query vector and sends it to the server.
  2. Still in the **online** phase, the server responds to this query by multiplying the query vector with the DB matrix.
  3. Finally, the client returns the result by decrypting the response using the preprocessed query parameters.

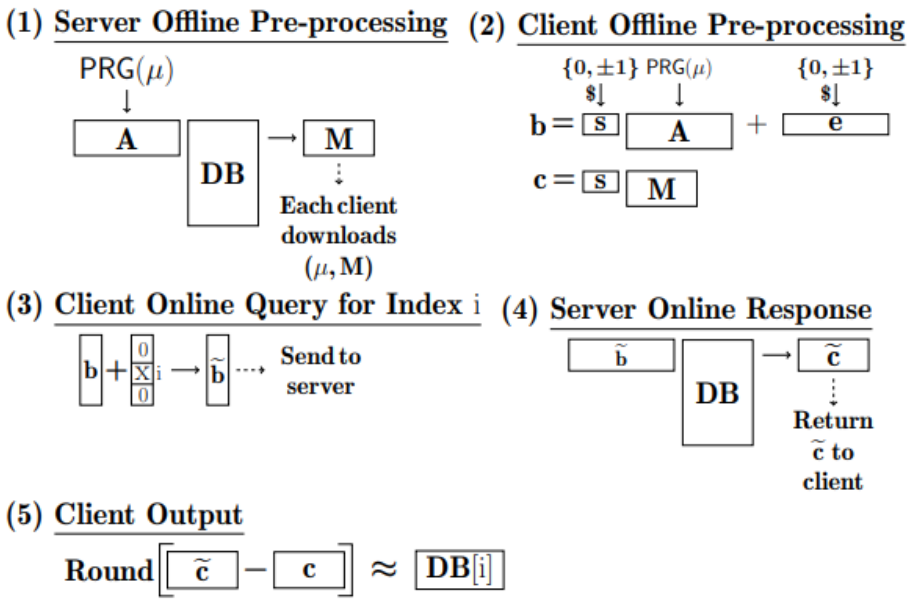

Fig. 1. An overview of FrodoPIR. In (1), the server compresses their database **DB** (represented as a matrix) into **M**, via multiplication with the global matrix **A** that is derived randomly from a public seed μ. The client downloads $(\mu, \mathbf{M})$, and in (2) they preprocess a query and store $(\mathbf{b},\mathbf{c})$, note that **b** is an LWE sample and is thus randomly distributed. In the online phase, in (3), the client creates their query by adding an indicator value x to the i^{th} vector entry of $\widetilde{\mathbf{b}}$. In (4), the server multiplies the client query vector with their DB matrix and return the result, $\widetilde{\mathbf{c}}$. Finally, in (5), the client subtracts **c** from $\widetilde{\mathbf{c}}$ — rounding the result to remove any error terms — and learns the i^{th} row of DB. The full scheme is given in Section 4.

Figure 1: FrodoPIR overview

5 Objectives

The goal of the project approach is to reduce the computational cost of the online query processing, allowing the client to deal with multiple indices simultaneously. By structuring the database as a $\sqrt{m} \times \sqrt{m}$ matrix D, each cell representing a different element in DB, the client sends then two query vectors v_{row} and v_{col}, each of size $\sqrt{m}$. The server then computes matrix-vector products to obtain the given entries.

6 Schedule

This is the scheduling estimation for this project:

Project Schedule	2023					2024								
	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
1. Study the theoretical background and review existing literature														
2. Choosing the project approach														
3. Creation of the project proposal														
4. Formalization of proposal														
5. Theoretical analysis of the model														
6. Model Implementation														
7. Result Evaluation														
8. Monograph Writing														
9. Presentation Elaboration and Finalizing														

Table 1: Capstone project schedule

6.1 Schedule details

1. Study the theoretical background and review existing literature
    - Reading theoretical background of Fully Homomorphic Encryption, Learning With Error problem, Ring Learning With Error problem.
    - Understandin the Bootstrapping approach.
    - Reading about most relevant cyphers for the context: DGHV and GSW.
    - Reading about Private Information Retrieval and most famous protocols (SealPIR, FrodoPIR, etc.)

2. Choosing the project approach
    - Deciding how we can contribute to the Private Information Retrieval current state of art.
3. Creation of the project proposal
    - Choosing the articles to be main reference.
    - Understanding the protocol chosen (FrodoPIR) and its mathematical implications.
    - Understanding the protocol scheme and mathematical overheads.
    - Finding how can we contribute to the protocol.
4. Formalization of proposal
    - Formalizing the project approach.
5. Theoretical analysis of the model
    - Start the formalization of the modifications.
    - Mathematically formalize the approach.
    - Do the cost analysis of the changes.
    - Do eventual changes of the approach in order to keep the correctness and viability.
6. Model Implementation
    - Understand the current implementation of the protocol
    - Select and analyse the parts to be modified.
    - Review the viability of the theoretical model according to the implementation
    - Do the hands-on of the theoretical model and change the current implementation.
7. Result Evaluation and Comparisions
    - Extract the results of the applied modifications
    - Compare with the original protocol.
    - Decide whether the optimization happened or not.
8. Monograph Writing
    - Compile the collected information.
    - Group and summarize according to the offered model of monograph.
    - (Let's do the writing!)
9. Presentation Elaboration and Finalizing
    - Do any necessary adjustments.
    - Elaborate poster.
    - Elaborate the presentation for the scheduled date (aprox. December).

References

[1] Asra Ali, Tancrède Lepoint, Sarvar Patel, Mariana Raykova, Phillipp Schoppmann, Karn Seth, and Kevin Yeo. Communication–computation trade-offs in pir. Cryptology ePrint Archive, Paper 2019/1483, 2019. https://eprint.iacr.org/2019/1483.

[2] Sebastian Angel, Hao Chen, Kim Laine, and Srinath Setty. Pir with compressed queries and amortized query processing. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 962–979, 2018.

[3] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 41–50, 1995.

[4] Henry Corrigan-Gibbs, Alexandra Henzinger, and Dmitry Kogan. Single-server private information retrieval with sublinear amortized time. Cryptology ePrint Archive, Paper 2022/081, 2022. https://eprint.iacr.org/2022/081.

[5] Alex Davidson, Gonçalo Pestana, and Sofía Celi. Frodopir: Simple, scalable, single-server private information retrieval. Cryptology ePrint Archive, Paper 2022/981, 2022. https://eprint.iacr.org/2022/981.

[6] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[7] Alexandra Henzinger, Matthew M. Hong, Henry Corrigan-Gibbs, Sarah Meiklejohn, and Vinod Vaikuntanathan. One server for the price of two: Simple and fast Single-Server private information retrieval. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3889–3905, Anaheim, CA, August 2023. USENIX Association.

[8] Muhammad Haris Mughees, Hao Chen, and Ling Ren. Onionpir: Response efficient single-server pir. Cryptology ePrint Archive, Paper 2021/1081, 2021. https://eprint.iacr.org/2021/1081.

[9] Sarvar Patel, Giuseppe Persiano, and Kevin Yeo. Private stateful information retrieval. Cryptology ePrint Archive, Paper 2018/1083, 2018. https://eprint.iacr.org/2018/1083.

[10] Radu Sion and Bogdan Carbunar. On the computational practicality of private information retrieval. 01 2007.

[11] 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, 2023. https://eprint.iacr.org/2023/452.