PIR – Private Information Retrieval

Adrian-Viorel ANDRIU
National Institute for Research & Development in Informatics - ICI Bucharest
adrian.andriu@ici.ro

Abstract: Before the private information retrieval (PIR) problem was recently identified, it was not thought possible to secure user privacy from a server.
A user can retrieve an item from a server that holds a database using the private information retrieval protocol without disclosing which item is retrieved. This article provides a motivation for the existence of the PIR protocol with examples from different fields of activity where only the PIR protocol helps. It also provides an overview of what already exists without concrete implementation details. The article concludes with future directions, more precisely with obtaining software applications based on the PIR protocol which has spread and is being used all over the globe.
Keywords: cryptography, database, private information retrieval, privacy, data security, PIR.

INTRODUCTION

At a theoretical level, private information retrieval (PIR) is a general problem regarding the action of private retrieval of the x-order bit from a string of bits stored on a server, a string of N bits. „Private" denotes that the server is unaware, that it cannot obtain or learn any information about x, i.e. the server does not know what is of interest to the person or entity that requested the information.

The need for online privacy increases as we become addicted to technology. Nowadays, knowledge about user preferences is information of well-recognized importance and value. This information is very important for companies and can play a crucial role in certain situations but sometimes to the detriment of the user. User preferences are known to be a secret that is part of the natural privacy of anyone but the server. This assumption, that the server will not use the user's tastes towards the user, was adopted as such and thus the trust in data manipulation systems was born because no system is ideal and sometimes trust is needed to be able to use software. However, there is no reason to make this assumption as it is possible to use the existing applications without trusting the server. One of the largest online media retailers stated that its database of millions of user profiles and shopping preferences is one of the company's assets.

Sometimes, for various technical and configuration reasons, the server can be honest but bad. This means that a server may be improperly configured and security breaches may occur. They offer the possibility to malicious people to manipulate the data, which theoretically, only the server was accessible

https://doi.org/10.54851/v4i2y202207

to authorized persons. Up to half of the top online servers are reported to compromise user privacy in this way. There are cases when some companies due to bankruptcy are forced to sell these databases of users' preferences. (Disabatino et al., 2000; Beaumont, 2000).

Nowadays, a user's preferences depend on the following aspects: the confidence that a certain company has globally, the quality of the infrastructure security services, the economic situation of the companies and their stability over time. These hypotheses, as well as others that are not mentioned, are too many to be true at the same time.

Solutions based on the PIR protocol would make possible the situation in which a user keeps his private preferences from everyone but absolutely everyone.

The remainder of this paper is structured as follows. Section 2 refers to the motivation for employing the PIR protocol. Section 3 includes a classification list for private information retrieval approaches, referring to what has been done in this field based on public scientific papers. Section 4 addresses certain open problems in this field. Section 5 provides some information about a simple PIR implementation, while section 6 presents the conclusion of this paper.

MOTIVATION

In this section I will provide examples of applications that use or can use PIR technology; I will also explain why certain approaches have failed and cannot be labeled as PIR solutions.

Examples

In the following I will describe concrete but also hypothetical examples in support of the utility of PIR.

A first example is related to patent databases. If a server in such an infrastructure knows what patent a user has been looking for, this can cause a lot of unpleasant situations for a researcher or inventor. Take for example the situation in which an individual discovers an idea, for example 2 + 2 = 4. Normally, he may want to patent this great idea. But first he has to consult an international patent database to see if something like this has been patented by someone else. Therefore, the question of that scientist can be accessed by the infrastructure admin and this aspect can generate information such as:

- A new invention. 2 + 2 = 4 can be patented. Why shouldn't I try?
- The administrator also has access to the scientist's area of activity.

This information should not be disclosed and is extremely important. In this situation, one solution would be PIR: the user could pay for the download of a single patent and the server and no one else will know which patent has just been downloaded.

Databases

Usually, pharmaceutical companies go either in the direction of drug inventories or they collect key information related to basic components, such as their properties (pharmaceutical databases).

These databases are used when more information is gathered about the basic components needed in the process of synthesizing a new drug. Thus, entire databases are bought by drug designers to hide a company's plans.

This involves high costs that could be avoided with the help of a PIR protocol, in this way the designers would buy only the necessary information.

Databases in various electronic fields

This category includes the commercial files of electronic publications, music files (mp3s), photos, videos, etc. It is well known that client data is placed under the trust of the server. In this case, user preferences may be hidden when purchasing digital products online. This highlights that users may be interested in the PIR protocol.

Examples from practice

A special operations department within the defense ministry plans an operation in a region R. This situation involves sending a request to the database of IT maps to obtain a high-resolution map of R. In this way, the IT can find out that a special operation is going to take place there. The question is whether the secret of a special operation within the department

can be kept secret and yet if it is possible to query an external database? The answer can be a positive one if PIR is used.

Isabelle Duchesnay suggests another hypothetical application. A spy has a corpus of different state secrets. Each secret has a catchy title, such as „Where's Abu Nidal?". He will not agree to reveal several secrets of two secrets in exchange for partial or full information disclosure. As potential buyers, you are reluctant to express the secret you want to obtain, as this could be sold further because it provides information about your interest (under the heading „who's looking for <TITLE>"). PIR offers a middle ground for both of you because you can take the secret in a private way.

Approaches and reasons why they did not work

Two direct approaches to the PIR problem were identified. In essence, they fail to solve the problem, but they offer us properties of potential practical solutions regarding PIR.

Scenario where the user can create a local copy of the database

The client can run queries on its local copy, which theoretically solves the PIR issue during the full database transfer process (from server to client). As a result, the server is unaware of the user's queries' contents and is therefore unaware of the client's preferences. Since the method is expensive and the user must pay for every database record, it has no real-world application. A transmission of the required information that is equal in size to the database's additional cost is incurred. Considering the cost of the database's overall contents, this expense is insignificant.

Use of anonymization techniques

If an anonymization technique is used, a user can send queries anonymously to a server and the responses it sends back to the user are also anonymous. In addition to these aspects, using in parallel an anonymous payment system (as in Gutmann, 2000) the user can pay anonymously for the execution of a query.

Based on the above information, we can conclude that this is a PIR solution. This is not true because the server can gather information and statistics about what the user was looking for.

It can also see which record has been accessed more times than others and many other statistics like this.

In this situation a significant disadvantage is that the anonymization techniques of the networks are:

- dependent on a third-party entity in which users are obliged to trust.
- insecure in the attack all against one; an attack in which several individuals cooperate against a single individual.

PRIVATE INFORMATION RETRIEVAL APPROACHES

The issue of PIR was first formulated in (Chor et al., 1995) and since then a large number of papers and articles have appeared in the scientific field. An important aspect is the fact that these works approach PIR from a theoretical perspective and the indications for practical implementation are few.

I classified these results according to what hypotheses were drawn by the authors in public scientific papers. I did not explain any algorithm due to space limitations and practical implementation issues but information was provided on some basic ideas of some algorithms.

Theoretical PIR

When I say theoretically I mean that the privacy of users falls under the assumption that this and confidentiality are undoubtedly independent of the computing power of a third-party entity. Chor in the paper (Chor et al., 1995) proved that a trivial solution is one in which the PIR solution has a communication with a lower limit equal to the size of the database. On the other hand, a high-performance solution is one that has less communication than the size of the database.

With this idea in mind, Chor et al. (1995) simplified the problem to some extent. They started from the hypothesis that there are several servers with the same data but which do not communicate with each other. This approach makes possible and feasible the idea of non-trivial PIR. The basic idea of the paper (Chor et al., 1995) is to send multiple queries to multiple databases.

These queries are constructed in such a way that they do not provide information about the object in which the user is interested. But, using the answers received after the requests, the user is able to get the desired answer.

There is a lot of information on this theoretical PIR topic in the paper of Chor et al. (1995).

Block PIR

One solution to the PIR problem is PIR of blocks. This approach assumes that database records are multi-bit blocks. The concept of theoretical PIR of blocks was first introduced in (Chor et al., 1995) and further analyzed. The PIR of blocks technique is crucial for making PIR practical.

Computational PIR

In order to obtain a better complexity of communication, another assumption was stated by Chor et al. (1995) in their paper. The term "computational" refers to the fact that the servers that hold the databases are connected by computing power. That is, under an appropriate inactivity assumption, the databases cannot obtain information about object of interest.

For every $e>0$, (Chor et al., 1995) presents a two-database Computational PIR scheme with communication complexity $O(N^{\varepsilon})$.

In the paper In the paper (Ostrovsky & Shoup, 2001) Ostrovsky and Shoup build PIR protocols with the option to write and record to the database in such a way that database servers do not know about the object of interest. There are protocols for both theoretical PIR and computational PIR with two or more servers. For example for theoretical PIR with several servers, we take a case with a number of three servers, they offer a protocol with complexity (N 1/3 log3 N).

Scenario with a single database and PIR technology

Here, it should be reminded that the first paper in which it was approached proved that the PIR problem has no non-trivial options for the case when a single database is used. Surprisingly, the substitute of a theoretical data protection with an assumption of intractability lets in the awareness of a non-trivial PIR protocol for a single database schema (Chor et al., 1995).

This is the first unique database protocol that designers adopted which guarantees database confidentiality. This is an improvement on the polynomial communication complexity in (Chor et al., 1995). This result is particularly effective because the user only needs to send at least N register bits to the database address and bit (the bit he wants to receive), regardless of protocol confidentiality.

Symmetric PIR

A PIR problem is related to symmetric PIR where database confidentiality is taken into account. This type of symmetric protocol must prevent a user from learning multiple recordings during a session. When working with billing processes, a very important property for practical applications is database confidentiality. The symmetric PIR for the case with a single server was considered in the paper of Gertner et al., (1998) for the first time and for the case with several servers it was approached in (Gertner et al., 1998).

PIR concept with hardware

In another scenario, Smith and Safford addressed in (Smith & Safford, 2000) the issue of the single PIR database considering the use of a special device against unwanted manipulation. In order to better understand this idea, it is assumed to use a secure coprocessor (Yee, 1994) which is attached to the server.

The client scrambles the query to be launched and sends it to the secure coprocessor for processing. The coprocessor decrypts the received query, then processes it and sends the answer encrypted.

The server has no proof of what the inquiry is, since:

- The server cannot access the RAM area of the attached coprocessor. This is one of the main properties. In other words, the server cannot see what the user-created queries look like.
- When the secure coprocessor receives a query, it will read all the records in the server's database so the record of interest is not revealed.

Further Extensions

Most of the scientific papers that have addressed the issue of PIR have focused on optimization at the communication level as this is the most expensive resource.

However, the applicability of such proposals remains debatable because their actual implementation encounters many problems (Beimel, Ishai & Malkin, 2000). The first problem concerning their implementation is that the computing power of servers increases in direct proportion to the size of the databases and the typical scenario is when the database contains a lot of data.

A solution in this regard has been proposed by (Y. Gertner, S. Goldwasser, and T. Malkin et al., 1998) which refers to the actual moving of calculations from servers where databases are kept on servers specifically dedicated to query processing.

Another solution is presented by Di-Crescenzo, Ishai & Ostrovsky (1998) in which special purpose servers are used for processing. This model processes the information offline, more precisely it is performed only once, regardless of the number of queries received. In both Di-Crescenzo, Ishai & Ostrovsky (1998) and Gertner et al. (1998) confidentiality is not protected if the servers communicate with each other against the user.

It was demonstrated that, while without any preprocessing linear computation is unavoidable, with pre-processing and some extra storage, computation can be reduced. Namely, Beimel et al. (2000) have the following results for the Theoretical PIR and any $k \geq 2$ and $E > 0$:

- A k-server protocol with $O(N^{1/(2k-1)})$ communication, $O(N/E\log^{2k-2}N)$ work, and $O(N^{1+E})$ extra storage bits.
- A k-server protocol with $O(N^{1/k+E})$ communication and work, and $O(N^{1+E})$ extra storage bits.

The targeted web advertising without revealing user preferences (a problem similar to PIR) is investigated in (Juels, 2001).

OPEN PROBLEMS

After a brief analysis of the existing results, we have drawn a series of open problems that can be considered in the following directions.

The idea of pre-processing and offline communication as well as the optimization of computing and online communication are important targets to be achieved in practical applications. We must understand that the work that has been done in these scientific papers has started from the idea that servers will not communicate against the user. PIR technology with a single database in which communication is done offline has been developed and discussed independently in scientific papers.

In practical applications and worldwide usage, the definition of the query can be summarized by giving me the x block from N records, but it needs to be developed in a more general form. Such aspects have been touched upon in the work of Chor & Gilboa (2000). In this sense I will give the following example: an individual has a certain DNA sequence and wants to look for similarities with external DNA samples in a database. In order to make comparisons privately and confidentially, classical PIR approaches are not good and better and more efficient algorithms are needed that can execute other types of queries than give me the x block from the N string.

Finally, earlier research ignored PIR techniques that are unique to particular applications. For instance, basic assumptions for digital libraries may be different from those for traditional databases.

SIMPLE THEORETICAL PIR IMPLEMENTATION

Let us consider the following scenario. Alice wants to obtain information from a database but does not want the database to learn what information she wanted. So one of the solutions is Information Theoretic PIR. But first I will explain this in a more general manner.

Assume one has 2 copies of the chosen database. For this practical example, where one can find the source code in Appendix A, I have tested a x-DB PIR scheme where one can view the database as an bit array and use the proprieties of XOR operation.

So, the number of the elements in the database will be filled with random 0 and 1 and an index will also be chosen randomly. This index i will be retrieved privately. I also saved the actual value at that index in order to make a verification at the end of the program. In order to make a matrix, I have used padding with the purpose of obtaining a perfect square.

These values won't be accessed anyway. The list should be split into lists and stored in two separate databases DB1 and DB2 that do not communicate.

Only the column indices with ‚1' in Z1 will be considered when XOR sum is taken across the columns, where Z1 is a random query made to DB1, and the final result is stored in A which is the column containing the element at index_i. The desired element is at index ‚row' in this list.

CONCLUSION

By using Private Information Retrieval protocols, the user has the chance to protect his privacy by hiding the items he had retrieved from the database. A comprehensive introduction to PIR was presented, focusing on potential applications, existing results and future work. With regard to possible future works, the design area of these PIR protocols should be targeted, as the theoretical variations have already been well researched.

REFERENCE LIST

Beaumont, C. (2000). *What price privacy when dotcoms go down?*. New Zealand Herald.

Beimel, A., Ishai, Y., & Malkin, T. (2000). Reducing the servers computation in private information retrieval: PIR with preprocessing. *Proceedings of CRYPTO 2000*, 55-73.

Chor, B., & Gilboa, N. (2000). Computationally private information retrieval. *Proceedings of the 29th annual ACM Symposium on Theory of Computing (STOC'97),* New York, 304–313.

Di-Crescenzo, G., Ishai, Y., & Ostrovsky, R. (1998). Universal service-providers for database private information retrieval. *Proceedings of 17th PODC*, 91-100.

Gertner, Y., Goldwasser, S., & Malkin, T. (1998). A Random Server Model for Private Information Retrieval. *International Workshop on Randomization and Approximation Techniques in Computer Science*, 200-217.

Gertner, Y., Ishai, Y., Kushilevitz, E., & Malkin, T. (2000). Protecting Data Privacy in Private Information Retrieval Schemes. *Journal of Computer and System Sciences*, *60*(3), 592-629.

Gutmann, P. (2000). An open-source cryptographic coprocessor. *9th USENIX Security Symposium Paper,* Denver, Colorado, USA, 97–112.

Juels, A. (2001). Targeted advertising... and privacy too. *CT-RSA 2001: Topics in Cryptology — CT-RSA 2001,* 408–424.

Smith, S. W., & Safford, D. (2000). P*ractical private information retrieval with secure coprocessors* (Technical report). IBM.

Yee, B. S. (1994). *Using Secure Coprocessors* (PhD thesis). CMU.