# PRIVACY PRESERVING INFORMATION RETRIEVAL OVER UNSYNCHRONIZED DATABASES

Meenu Poulose[1] and Tinku Soman Jacob[2]

[1] Student, Department of Computer Science and Engineering, MBITS, Nellimattom.
[2]Assistant Professor, Department of Computer Science and Engineering, MBITS, Nellimattom

## *ABSTRACT*

*A database is a collection of information that is organized so that it can easily be accessed, managed and updated. It may also contain the private and public information about an individual. The user access the information by giving some relevant queries. When a user retrieves any data the server may able to identify which the data is. If the user accesses the $x^{th}$ data then that 'x' must be hidden from the server. Private information retrieval can solve this issue. Here introducing an efficient PIR scheme that uses multiple servers. Each non-colluding server stores the identical copies of the database. When any of the servers contain non-identical copy of database i.e. unsynchronized database the user will get an error. Our proposed system works properly even under such circumstances since it has the mechanism to find the unsynchronized databases. This multi server PIR scheme has the same computational and communication complexity. It also allows multiple PIR queries simultaneously.*

## *KEYWORDS*

*Private information retrieval, information theoretic privacy, database management, distributed source coding.*

## 1. INTRODUCTION

Most of the web services require the user sign up by giving some personal information like mail id, date of birth etc. But these information are not much safe within those servers. It may be viewed by some other third parties also. For example we have an account in an online shopping site and we used to view a particular product that we want to buy. When we visit any other site some pop-up messages may came. It may be an advertisement from the previous site about our favourite product. Here the identity and the retrieved data are disclosed by the server. It demands the need of private information retrieval.

We have many searching techniques that hide the user identity. But it is not a well protection since the contents we are searching is not hidden. By analyzing this contents can infer a conclusion about the user identity. To address this concern, some private search tools instead mask the contents of web queries. These tools include chaffing and winnowing approaches like TrackMeNot, encrypted database searches, private information retrieval (PIR), oblivious transfer,

DOI: 10.5121/ijci.2016.5226 

and private stream searches (PSS). Chaffing and winnowing involves sending bogus queries to a server and extracting only the relevant results. Private information retrieval (PIR) is a way for a client to look up information in an online database without letting the database servers learns the query terms or responses. A simple and inefficient way to do this is for the database server to send a copy of the entire database to the client, and let the client look up the information for herself. This is called trivial download. The goal of PIR is to transmit less data while still protecting the privacy of the query.

PIR protocols can be grouped into two classes corresponding to the security guarantees they provide. One class is computational PIR, in which the database servers can learn the client's query if they can apply sufficient computational power to break a particular cryptographic system. The other class of protocols thoseconsider in this work is information- theoretic PIR , in which no amount of computation will allow the reconstruction of the client's query. In these protocols, the query is protected by splitting it among multiple database servers. As is common in many distributed privacy-enhancing technologies, such as mix networks, Tor , or some forms of electronic voting , we must assume that some fraction of the servers above some threshold are not colluding against the client. It seems unlikely that multiple large-scale servers would maintain identical databases without colluding. Furthermore, there is little incentive for existing service providers like Google or Yahoo to enable private searches.

In this paper we first introduce the basic multi server PIR system with synchronized databases and then explain the proposed system i.e. the PIR over unsynchronized databases. The key idea of our scheme is simple: here first determine which records are unsynchronized, and then construct a PIR query that avoids these problematic records. When the number of unsynchronized database records scales sub linearly in the database size, our scheme has asymptotic communication and online computation costs that are identical to state-of-the-art PIR schemes. In practice, the system incurs slightly higher communication and server- side computation compared to traditional PIR. Our approach also allows multiplequeries to be processed in a single batch of PIR, unlike existing schemes.

## 2. RELATED WORKS

Since 1995, much work has been done creating protocols for private information retrieval (PIR). Many variants of the basic PIR model have been proposed, including such modifications as computational vs. information-theoretic privacy protection, correctness in the face of servers that fail to respond or that responds incorrectly, and protection of sensitive data against the database servers themselves.We begin with an example of information-theoretic, multi-server PIR proposed by Chor et al [2].

### 2.1. BASIC PIR SCHEME

Two servers store identical copies of a database of records $f = [f_1 \ldots f_n]^T$ , and a client wishes to retrieve the $w^{th}$ record, $f_w$. In practice, records can be of arbitrary length, but for simplicity, suppose each database element is a single bit, 0 or 1. The user's request can be represented by $e_w \in \{0,1\}^N$, the indicator vector with a 1 at index w and 0's elsewhere. To disguise this query, the user generates a random string $\alpha \in \{0,1\}^N$ with each entry a Bernoulli (1/2) random variable. The queries sent to servers 1 and 2 are $\alpha \oplus e_w$ and $\alpha$, respectively. Each server computes the inner product of its received query vector with the database x using bitwise addition (XOR) and returns

a single-bit result. The user XOR the results from the servers to get $f_w$ precisely. The scheme is illustrated in Figure 1. In an honest-but-curious adversarial model, this PIR scheme is information theoretically private, since received queries α and $\alpha \oplus e_w$ appear random.

### 2.2. COLLUSION RESISTANCE

Multi server PIR scheme will fail when the servers were colluded. The client can improve her privacy by querying many randomly-selected servers (e.g. in a P2P network); this reduces the likelihood of sending queries to colluding servers. Algorithmically, there exist PIR schemes that offer the property of κ-collusion-resistance as long as no more than κ of the d servers collude, information-theoretic security is guaranteed.

### 2.3. DISTRIBUTED SOURCE CODING

Distributed source coding (DSC) is an important problem in information theory and communication. DSC problems regard the compression of multiple correlated information sources that do not communicate with each other. By modelling the correlation between multiple sources at the decoder side together with channel codes DSC is able to shift the computational complexity from encoder side to decoder side. In our problem, the client is the receiver, and the servers are the distributed sources. Assume that the number of unsynchronized database elements s is small, so the servers' contents are highly correlated. The client must learn which database elements are unsynchronized. the difference of the servers' data - to successfully complete PIR.

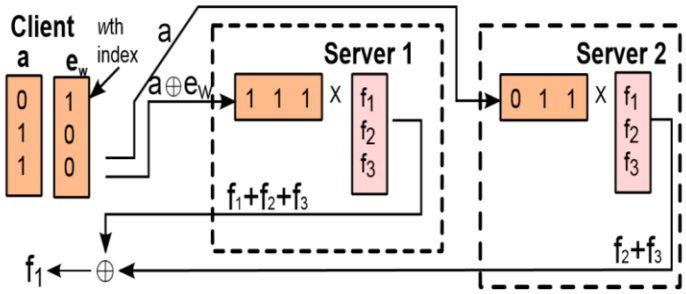


Figure 1: Basic two-server PIR scheme. Each server computes the bitwise sum of a user-specified subset of database records. Because the two user-specified subsets differ only at the $w^{th}$ index, the binary addition of each server's results gives the desired record.

## 3. PROBLEM DEFINITION

Existing PIR schemes do not have the mechanism to handle the unsynchronized databases. PIR allows a user to retrieve information from a database without revealing the server which data is retrieved. To achieve this privacy mainly two types of PIR schemes were introduced. Among them multi-server PIR is capable to achieve that privacy without more computations. Here the main requirement is that it needs multiple non-colluding servers. The requirements for multi-server PIR are;

1) Multiple servers are available.
2) Each server stores a duplicate copy of the database.
3) The individual servers do not collude.
4) The servers are honest-but-curious.
5) Servers willingly implement PIR algorithms.

Existing schemes only addressed the assumptions 3 and 4. Some PIR schemes are robust to Byzantine servers that return arbitrary , incorrect information [10]. Other schemes allow up to k servers to collude without losing any privacy [8].If any of the servers database is unsynchronized then the existing schemes will either produce an error or output an incorrect result. Figure 2 shows such a scenario. In order to handle such unsynchronized databases here introducing an efficient PIR scheme that can locate the unsynchronized databases and after modification the user can retrieve the desired data.

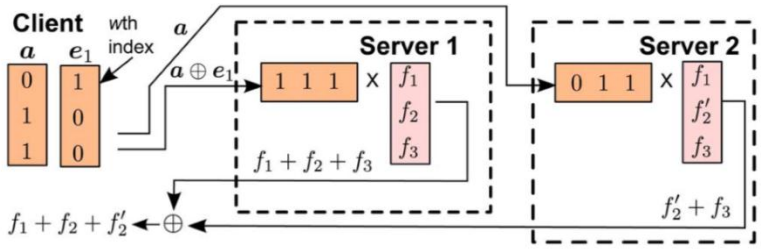


Figure 2: Basic two-server PIR scheme when one of the records contains an outdated data.

## 4. PRIVACY PRESERVING INFORMATION RETRIEVAL OVER UNSYNCHRONIZED DATABASES

Multi-server PIR schemes require multiple non-colluding servers. If any of the server collude to another then the privacy cannot be guaranteed. If we have strictly non-colluding servers still there exist a problem i.e. the unsynchronized databases. Consider the scenario, that we have three non-colluding servers and each server posses the similar copies of databases. But in the third sever the user failed to update his records. So it has an out-of-date entry. When the user requests for a data he will get the sum of the desired records with an error term. Some alternative methods are there to solve this problem. One is to treat the reply from the third server as an error and query the second or third server [5]. But it will increase the cost of communication since it needs to communicate with a server that is not already connected. The main aim here is to find the perfectly synchronized set of servers. Our proposed system asymptotically has the same computational and communication complexity as state-of-the-art PIR schemes for synchronized databases; this comes at the expense of probabilistic success and two rounds of communication (most existing schemes require only one).

The proposed scheme consists of two phases. Phase 1 find which records are unsynchronized and phase 2 retrieves the desired data to the user. The collusion-resistance scheme is used here for synchronized databases.

## 4.1. PHASE 1: LOCATE UNSYNCHRONIZED RECORDS

In the multi-server PIR schemes each server will have the same copies of databases. If any of the servers contains a mis-synchronized database we will find the location of that record in this phase. Unlike [4] to find the location this scheme relies on hash functions. Here assume that the hashes of each record are stored within the databases. Suppose one server posses the record $f_i$ and the other non colluding server also has the same record as $f_i$. Then both the servers will have same hash value H ($f_i$). If one of these two servers has an out-of-date record $f_i$ then both hash values will be different.

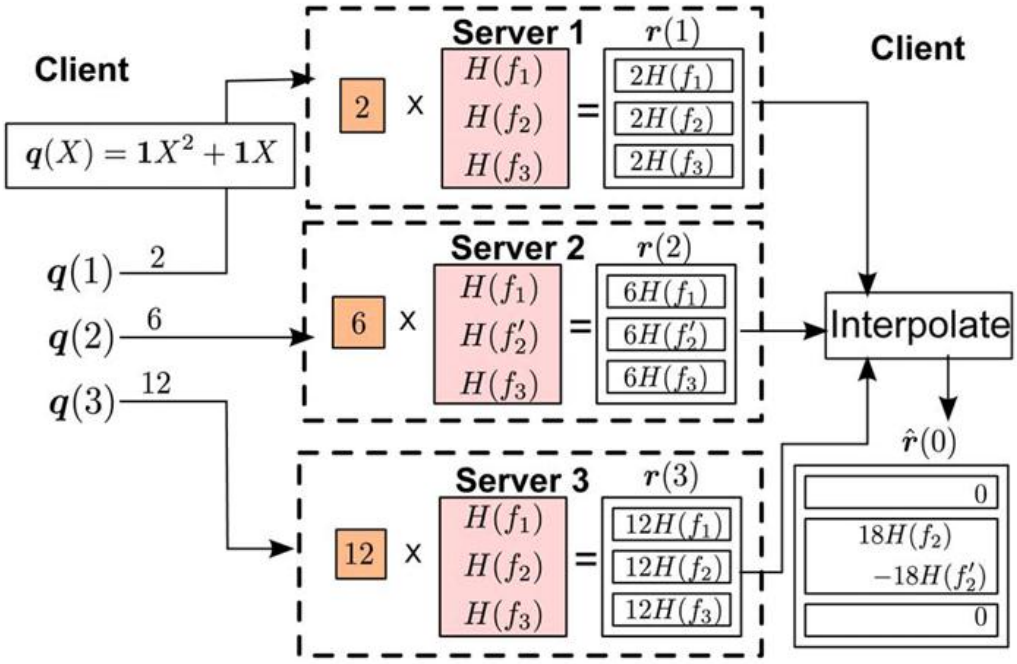


Figure 3: Identifying the location of unsynchronized records. Here $f_2$ is not synchronized across all servers.

Suppose we have only two servers, and a genie sums the servers' respective views of the database hashes over $GF(2^l)$, giving $H(f^{(1)}) + H(f^{(2)})$. The synchronized (i.e., equal) hashes cancel, giving a sparse vector of length $n$, with nonzero entries at the unsynchronized records. A parity check matrix could be used to compress this sparse vector for transmission to the client: $\mathbf{A}\cdot( H(f^{(1)}) + H(f^{(2)}) )$ By linearity, this is equivalent to the equation $\mathbf{A}\cdot H(f^{(1)}) + \mathbf{A}\cdot H(f^{(2)})$. So to communicate the same information in a distributed fashion, each server can simply compress its own database with a pre-determined $\mathbf{A}$ matrix, and the client can recover the sparse vector from the compressed vectors. This is for a two server architecture and the same idea works for more than two servers i.e. each server $S_i$ individually compresses its view the database by returning $\mathbf{A} \cdot f^{(i)}$ to the client. After this the client can do pair wise reconstruction finding the set of unsynchronized records.

## 4.2. PHASE 2: RETRIEVE THE DESIRED RECORD(S)

After the first round of communication the client knows the locations of the unsynchronized records. So the servers must ensure that they avoid touching those records. If the record $w$ is

unsynchronized then both server's query vectors should be zero at the $w^{th}$ index i.e. server neither touches the unsynchronized records. The same idea holds for more servers. Here we are not focusing on the data synchronization. Instead of it, this mechanism provides information retrieval even under the circumstance where the data in each server are not synchronized.

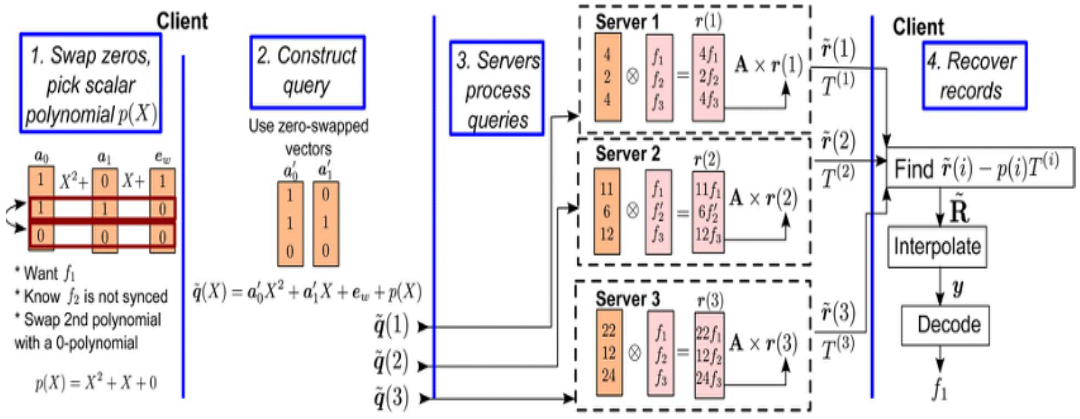


Figure 4: An example for PIR scheme, phase 2.Here the database have 3 records i.e. n=3

### 4.3. SYNCHRONIZATION

The existing multi-server PIR scheme describes the information retrieval process over unsynchronized database, but it doesn't have a phase to synchronize the outdated database. After retrieving the data there should be some steps to achieve synchronization. For that we can use the information that received after the phase 1. Phase 1 locates where the unsynchronized data records are. Hash functions of each record from each servers can be computed from the phase 1. After comparing the hash values from servers we can identify the unsynchronized server as well as thesynchronized servers. To achieve synchronization just replace the unsynchronized data with the data from synchronized records.

The required amount of communication and computation are similar to the other existing schemes. The main communication overhead in our scheme when compared to an optimal private information retrieval [3] is the downlink communication in phase 1 when locating unsynchronized database records. But the total communication complexity is same since the above overhead is dominated by the uplink in phase 1. So the communication complexity is same between our schema and [3]. Next is computation, our scheme requires the pre-computation of hash values of each record for the synchronization phase. Here the server computation cost is essentially equal to the [3], But client takes some additional computation to find the unsynchronized record's locations and to interpolate the desired records.

To understand the performance of this multi server PIR we can consider two factors i.e. the probability of success and the total query run time. The probability of success relies on the communication. The probability of success can be described as the probability of correctly retrieving the desired record. It increases as a function of communication cost. Some patters of query records and mis-synchronizations may generate decoding errors. That will result a client

being not recovering the requested record. Even when the number of mis-synchronized records is small this PIR scheme returns the desired records with a high probability.

Next is runtime, multi-server PIR requires more time to locate unsynchronized records when compared to other schemes like [3]. Runtime can be expressed as a function of the unsynchronized database records. Our runtime overhead is comparatively small, but it increases with the number of unsynchronized records. The overhead of our scheme does not increase as a function of database size, because the runtime overhead is dominated by locating the synchronization errors. Locating this is a process that depends on the number of unsynchronized records than the database size.

## 5. CONCLUSIONS

The privacy preserving information retrieval uses the multi server PIR scheme that can work even with the unsynchronized databases. This PIR scheme finds the correct location of the unsynchronized records and then retrieves the desired records. This scheme uses the same computational and communication complexities of the existing multi server PIR. It uses the distributed source coding to achieve the privacy. The proposed method needs the number of unsynchronized records to be small. And it is the first multi server scheme that returns the desired record even when server's databases are not perfectly synchronized.

## REFERENCES

[1] Giulia Fanti, Kannan Ramchandran, "Efficient Private Information Retrieval Over Unsynchronized Databases," Ieee Journal Of Selected Topics In Signal Processing, Vol. 9, No. 7, October 2015.

[2] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in Proc. IEEE FOCS, Milwaukee, WI, USA, 1995, pp. 41–50.

[3] C. Devet, I. Goldberg, and N. Heninger, "Optimally robust private information retrieval,"IACRCryptologye PrintArchive, vol.2012,p.83, 2012.

[4] G. Fanti and K. Ramchandran, "Efficient private information retrieval over unsynchronized databases," in Proc. Allerton, 2014.

[5] I. Goldberg, "Improving the robustness of private information retrieval," inProc. IEEESymp.Securityand Privacy, 2007,pp.131–148.

[6] D. Gross, "Yahoo Hacked, 450,000 Passwords Posted Online," CNN Tech., Retrieved from [Online]. Available: http://www.cnn.com/2012/ 07/12/tech/web/yahoo-users-hacked

[7] R.Sion and B. Carbunar,"On the computational practicality of private information retrieval," in Proc. NDSS, 2007, pp. 2006–2016.

[8] A. Beimel, Y. Ishai, and E. Kushilevitz, "General constructions for information-theoretic private information retrieval," J. Comput. Syst. Sci., vol. 71, no. 2, pp. 213–247, 2005.

[9] M. Barbaro and T. Zeller, A Face is Exposed for AOL Searcher no. 4417749 NewYorkTimes, Aug.2006 [Online]. Available:http:// www.nytimes.com/ 2006/08/09/technology/09aol.html

[10] A. Beimel and Y. Stahl, "Robust information-theoretic private information retrieval," in Proc. Security Commun. Netw., 2003, pp. 326–341, Springer.

## AUTHORS

Meenu Poulose is currently pursuing M.Tech in Cyber Security in MBITS, Nellimattom. She completed her B.Tech from MG University College of Engineering, Muttom, Kerala, India. Her area of specialization is Cyber Security.

Tinku Soman Jacob is currently working as the Assistant Professor in Department of Computer Science and Engineering at MBITS, Nellimattom, Kerala, India. He received his B.Tech Degree in Computer Science and Engineering from Mar Athanasius College of Engineering Kothamangalam and M.E in Software Engineering from Sree Krishna College of Engineering and Technology, Coimbatore. His area of specialization is Software Engineering.