Implement a Secure Method of Privacy-conscious Approach on SQL Query using bb-PIR

Zon Nyein Nway, Nan Sai Moon Kham
University Of Computer Studies, Yangon, Myanmar
zonnyeinnway7th@gmail.com, moonkhamucsy@gmail.com

Abstract

Private retrieval of public data is important when a client wants to query a public data service without revealing the query to the server. So, we want to use a practical and flexible approach for the private retrieval of public data called bounding box private information retrieval (bb-PIR). The bb-PIR can make a tradeoff between a cost of retrieval and the degree of privacy. And bb-PIR can provide secure data communication between a client and the server. Therefore, there is no doubt that bb-PIR is more practical because of its lower communication cost. This paper will report how Private Information Retrieval (PIR) can help users keep their sensitive information in an SQL query from being leaked. We will also show how to retrieve data from a relational database with bb-PIR by hiding sensitive constants contained in the SQL query.

Keywords: *Bounding Box Private Information Retrieval (bb-PIR), Private Information retrieval (PIR), Structured Query Language (SQL)*

1. Introduction

Most software system request sensitive information from users to construct a query, but user is unwilling to provide such information. The solution by private information retrieval (PIR) [1, 4] is to provide such user to retrieve data from a database without the database or the database administrator learning any information about the particular item that was retrieved. Practical PIR scheme is important to maintain user privacy in some application domains like patent databases, pharmaceutical databases, online census, real-time stock quotes, location-based services, and Internet domain registration. For instance, the current process for Internet domain name registration requires a user to first disclose the name for the new domain to an Internet domain registrar. Subsequently, the registrar could then use this inside information to preemptively register the new domain and thereby deprive the user of the registration privilege of that domain. Therefore, many users find it unacceptable to disclose the sensitive information contained in their queries by the simple act of querying a server.

Although the privacy of users' in querying information from the database server is protected well by keyword-based PIR scheme, the privacy of querying valuable information with bb-PIR scheme is not done yet. By using bb-PIR; a client can retrieve a data string from a database without the index of the data string being revealed to the database. And since relational databases and SQL are the most influential of all database models and query languages, we argue that many realistic systems needing query privacy protection will find our approach quite useful.

2. Motivation

Privacy is important issue today. It is the ability of a person to control the availability of information about him or herself. Privacy categories of special concerns are (1) Political privacy (2) Consumer Privacy (3) Medical privacy (4) Information technology end-user privacy; also called data privacy (5) Private property.

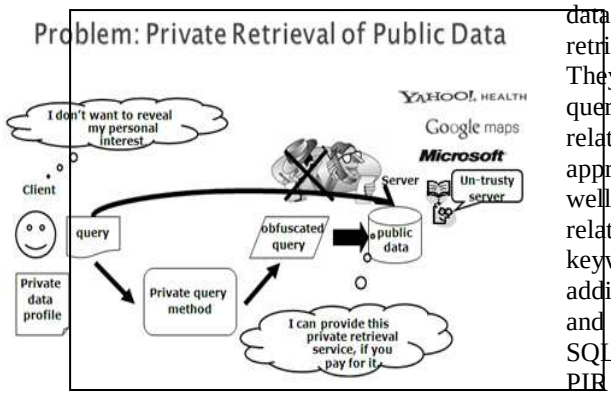

Figure.1. Problem of Private Retrieval of Public Data

Data Privacy problems (in Figure .1.) exist whenever uniquely identifiable data relating to person is collected and stored in digital form. Improper control can be the main cause for privacy issues. The most common sources of data that are affected by data privacy issues are health information, criminal justice financial information and genetic information. The challenge in data privacy is to share data while protecting the personally identifiable information.

3. Related Work

Wong, Agrawal and Abbadi proposed a Bounding Box PIR (bb-PIR) that unifies both k-Anonymity and cPIR. A client anonymizes her private query data in a rectangle called bounding box, whose range corresponds to a sub matrix of public data matrix. The size of the bounding box is determined by the client's privacy requirement and desired charge limit. The area of the bounding box determines the privacy that the client can achieve, the larger the area, the higher the privacy obtained, but with higher computation and communication costs, vice versa. [5]

Olumofin and Goldberg addressed the problem of preserving the privacy of sensitive information within an SQL query using PIR. They solve two obstacles to deploying successful PIR-based systems. First they develop a generic data access model for private information retrieval from a relational database using SQL. They show how to hide sensitive data within a query and how to use PIR to retrieve data from a relational database. Second, they develop and approach for embedding PIR schemes into the well-established context and organization of relational database systems. They extended keyword based PIR to B+ tree and PHF. In addition, they provide and implemented system and combine the technique with the expressive SQL. We intend to replace the key-word based PIR to bb-PIR [13].

We want to implement how to retrieve data from relational database by hiding sensitive constants contained in the SQL query.

4. Private Information Retrieval

PIR provides a means to retrieve data from a database without revealing any information about which item is retrieved. In its simplest form, the database stored a n-bit string X, organized as r data blocks, each of size b bits. The user's private input or query is an index is an index $i \in \{1, r\}$ representing the i^{th} data block. A trivial solution for PIR is for the database to send all r blocks to the user and have the user select the block of interest at index i (i. e., X_i) ;but this carries a very poor communication complexity.

The three important requirements for any PIR scheme are correctness (return the correct block X_i to the user), privacy (leaks no information to the database about i and $X_{i)}$ and non-triviality (communication complexity is sub-liner in n) [17].

4.1. K-anonymity based PIR

K-Anonymity has been used in privacy-preserving location-based services [14], where the location point of a user is blurred into a cloaked region consisting of at least k nearby user locations and the server returns the nearest point of interests to the cloaked region. The parameter k serves as a configurable degree of privacy. Similarly in the more general setting of private retrieval, one could insert into the private

query some random data that is close to the private data in the query, such that a private data item cannot be identified from at least k data items. Then the server returns all the public data that matches the anonymized private data. However, a potential security threat with k-Anonymity is that both the client query and the server answer, although anonymized for protecting the client's privacy, are in plain text that can be seen by a third party. The privacy of k-Anonymity for numeric data has also been questioned by a number of proposals [15, 9, and 10] for potential proximity breach: the real private data and the blurred data could be so close that the server can conclude with probability 1/k that the private data is in a narrow range.

Let RT (A_1, .A_n) is a table, QI_{RT}= (A_i,…., Aj) be the quasi-identifier associated with RT, A_i,…, Aj $\subseteq A_1$,….,A_n and RT satisfy k-anonymity. Then, each sequence of values in RT [A_x] appears with at least k occurrences in RT [QI_{RT}] for x=i,…..,j.

4.2. Computational PIR (cPIR)

Computational Private Information retrieval (cPIR) [8] retrieves a bit from a public bit string on a server without revealing to the server the position of the desired bit under some intractability assumption. To achieve the most balanced performance for communication and computation costs, the cPIR protocol requires the public data to be organized as a matrix. It achieves computationally complete privacy by incurring expensive computations over all public data on the server, and keeps the data communication secure by transmitting random information hiding vectors. The exposed, chargeable data is only a column of the public data matrix. Due to its expensive computation costs on the server, even the cPIR technique with the least expensive operation, modular multiplication [8] is criticized as being up to two orders of magnitude less efficient than simply transferring the entire data from the server to the client [15].

4.3. Bounding Box PIR (bb-PIR)

To make a trade-off between the cost of retrieval and the degree of privacy, this system used a generalized private retrieval approach called Bounding-Box PIR (bb-PIR) that unifies both k-Anonymity and cPIR. The public data is organized as a matrix as in cPIR. A client anonymizes her private query data in a rectangle called bounding box, whose range corresponds to a sub-matrix of the public data matrix. The size of the bounding box is determined by the client's privacy requirement. The area of the bounding box determines the privacy that the client can achieve, the larger the area, the higher the privacy obtained, but with higher computation and communication costs, and vice versa.

BbPIR is similar to *c*PIR as it retrieves one bit at a time. In order to retrieve a *b* bit item *x*, *bb*PIR can be repeated *b* times. The client query, row vector *y*, can be reused *b* times on *b* bits of *x*. Only the server answer, column vector *z*, needs to be re-calculated for each of the *b* bits.

The *bb*PIR protocol is described as follows:

1. Initially, the client sends to the server an m-bit number N which is the product of two random m/2-bit primes p1 and p2, and the dimensions of the bounding box BB of area 1/ρ. The number of rows and columns, r and c in the bounding box BB are decided as follows:
If $\mu \geq \lceil\sqrt{1/(\rho.b)}\rceil$, set

$$r = \lceil\sqrt{1/(\rho.b)}\rceil, c = \left\lceil\sqrt{b/\rho}\right\rceil$$

Otherwise, set r = min (μ, $\lceil 1/\rho \rceil$, s)

$$c = \min(\lceil 1/(\rho.r)\rceil, t)$$

2. To retrieve entry (e, g) in M, the client first places BB on M with the above defined dimensions r, c, s.t. BB covers (e, g), and BB is within the address space of M.

3. The client generates a vector of *c m*-bit random numbers in Z_N^{+1}, $y = [y_1, ..., y_c]$, *s.t.* y_g *is a QNR(* Quadratic non residuosity*) and all other*

y_i *(i = g) are QR* (Quadratic residuosity*)*. It sends the coordinates of *BB and* vector y to the server.

4. The server computes for each row *i of* the sub-matrix BB a modular product z_i = , *where* $w_{i,j}$ = *if* $M_{i,j} = 0$,*and* $w_{i,j} = y_j$ *if* $M_{i,j} = 1$.

5. The server sends to the client $z_1, ..., z_r$.

6. The client determines that $M_{e,g} = 0$ *if* z_e *is a QR, and* $M_{e,g} = 1$ *if* z_e *is a* QNR.

7. Repeat steps 4-6 to obtain the remaining bits of the requested item in *(e, g).*

The following figure illustrates the example query, retrieving $M_{2,3}$ from a 4 × 4 bit matrix M. The placement of the bounding box BB is flexible, as long as it covers $M_{2,3}$. Because the sizes of vectors y and z are reduced, the computation and communication costs are reduced proportionally.

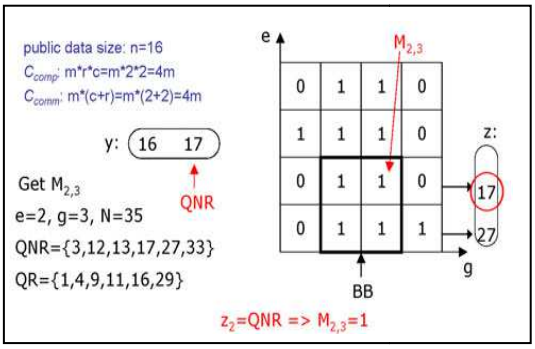

Fig.2. *Bb*-PIR Example: Private Retrieval of One Bit

4.2.1. Quadratic Residues

Quadratic Reciprocity relates the solvability of the congruence $x_2 = p \pmod q$ to the solvability of the congruence $x_2 = q \pmod p$, where p and q are distinct odd primes.

If p is an odd prime, there are equal numbers of quadratic residues and quadratic nonresidues among {1, 2, . . . , p − 1}.

If p is an odd prime, a > 0, and (a, p) =1, the Legendre symbol – is defined by

–

5. Proposed System Scheme

The proposed scheme consists of four components-Client, PIR Manager, Relational Database, and Indexed Table.

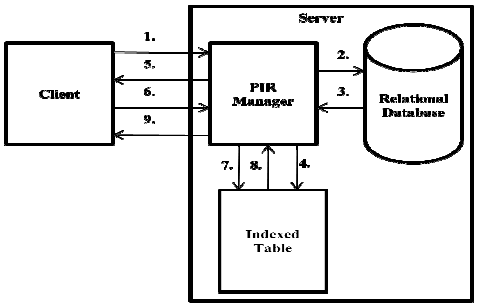

Figure.3.The Framework of the Proposed System

The flow of the proposed scheme is depicted in Figure. 3. The server consists of PIR Manager, relational Database and Indexed Table. We describe our system with an example.

```
SELECT    t1.name, t1.email, t2.date
FROM      user t1, contact t2
WHERE     (t1.user-id=t2.user-id) AND
          (t2.date>20100101)    AND
          (t2.damain="jaju.com")
```

Figure.4. Example Query with a WHERE clause with sensitive constants

Step (1): To protect private information, it is necessary to know which information is private. In this step, contains the extraction of sensitive constants in SQL Query and the hiding of these sensitive constants. We do hiding sensitive constants in the predicates of the WHERE clause in figure.4. The process for the above SELECT query is stated below.We assume the "**20100101**" and "**jaju.com**" are private.

The original query is divided into sub-queries containing sensitive data. Initially, the client builds an attribute list, a constraint list, and a sub-SELECT query, using the attribute names and the WHERE conditions of the input query.

To begin, initialize the attribute list with the attribute names in the query's SELECT clause, the constraint list to be empty, and the subquery to the SELECT and FROM clauses of the original query.

– *Attribute list: {*t1.name, t1.email, t2.date*}*
– *Constraint list: {}*
– *Subquery:* SELECT t1.name, t1.email, t2.date FROM user t1, contact t2

Next, consider each WHERE condition in turn. If a condition contains a private constant, then add the attribute name to the *attribute list* (if not already in the list), and add (attribute name, constant value, operator) to the *constraint list*. Otherwise, add the condition to the subquery.

On completing the above steps, the attribute list, constraint list, and subquery with reduced conditions for the input query become:

– *Attribute list: {t1.name,* t1.email, t2.date, t2.domain*}*
– *Constraint list: {*(t2.date, 20100101, >) AND (t2.domain,"jaju.com", =)*}*
– *Subquery:* SELECT t1.name, t1.email, t2.date FROM user t1, contact t2 WHERE (t1.user-id= t2.user-id)

The client sends this sub-query, a key attribute name and an index file type to the server. The key attribute name is selected from the attribute names in the constraint list-t2.date and t2.domain in our example. The choice may either be random, made by the application designer, or determined by a client optimizer component with some domain knowledge so that we will get an optimized choice.

Step (2): The PIR manager: executes the subquery on its relational database by bb-PIR method and the relational database returns the sub-query result to the server by step (3). The PIR manger generates a cached index of the specified type on the subquery result in step (4), and returns metadata for searching the indices to the client in step (5). Then the client performs one or more bb-PIR queries by step (6) and builds the desired query result from the data retrieved from indexed table rather than the original database by step (8).

Finally, the client combines the partial results obtained from the queries with set operations (union, intersection), and performs local filtering on the combined result, using private constant values for any remaining conditions in the constraint list to compute the final query result. The client thus needs query-optimization capabilities in addition to the regular query optimization performed by the server.

5.1. Indexed Hashed Table

For this indexed table, use hash indices of key attributes that are required to choose by the user or application. The server first computes the perfect hash functions for the key attribute values, and then inserts each tuple into a hash table. The metadata that is returned to the client for hash-based indices consists of the PHF parameters, the count of tuples in the hash table, and some PIR- specific initialization parameters.

The important benefits of this approach are the optimizations realizable from having the database execute the non-private sub-query, and the fewer number of bb-PIR operations required to retrieve the data of interest. In addition, the bb-PIR operations are performed against a cached index which will usually be smaller than the complete database [17].

6. Experiment

Compared to k-Anonymity, bb-PIR is secure in data communication between a client and the server, because it transfers the information hidden in a bounding box instead of plain text data. Moreover, bbPIR does not suffer from proximity breach as much as k-Anonymity, because the bounding box includes data values that are not close to the query value.

Compared to cPIR, bbPIR is more

practical because of its lower computation cost. BbPIR degenerates into k-Anonymity if the range of the bounding box is a single column on the public data matrix. At the other extreme, bbPIR becomes cPIR if the range of the bounding box is the entire public data matrix [17].

7. Conclusion

We present a privacy mechanism that uses bounding box private information retrieval to preserve the privacy of sensitive constants in an SQL Query. That use techniques to hide sensitive constants found in the predicates of the SQL Query, and retrieve data from indexed hash table rather than the original database. We hope that our work will provide valuable insight on how to preserve the privacy of sensitive information for many existing and future database applications.

References

[1] Beimel, A., Stahl, Y.: Robust Information-Theoretic Private Information Retrieval. J. Crypto. 20(3), 295-321 (2007)

[2] Bethencourt, J., Song, D., Waters, B.: New Techniques for Private Stream Searching, ACM Trans, Inf. Syst. Secure. 12(3), 1-32 (2009)

[3] Chan, C-H., Hsu, C-H.: Private Information Retrieval Scheme with E- Payment in Querying Valuable Information. Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference, IEEExplore.ieee.org (December 2009)

[4] Chor, B.k Goldreich. O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: FOCS, October 1995, pp. 41-50 (1995)

[5] Crescenzo, G.D.: Towards Practical Private Information retrieval. In: Achieving Practical Private Information Retrieval (Panel@ Securecomm 2006) (August 2006)

[6] Domingo-Ferrer, J., Bras-Amoros, M., Manjon, J.: User-private Information Retrieval Based on a Peer-to-peer Community. Data and Knowledge Engineering 68(2009) 1237-1252 Running (February 2008)

[7] ICANN Security and Stability Advisory Committee (SSAC). Report on Domain Name Front

[8] Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: FOCS, pp. 364–373 (1997)

[9] LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Workload-aware anonymization. In: KDD, pp. 277–286 (2006)

[10] Li, J., Tao, Y., Xiao, X.: Preservation of proximity privacy in publishing numerical sensitive data. In: SIGMOD Conference, pp. 473–486 (2008)

[11] Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: A privacy-aware location based database server. In: ICDE, pp. 1499–1500 (2007)

[12] Nakamura, H., Fujii, A., Ohtake, G.., Majima, K., Imaizumi, H., Fujita, Y., NHK, Tanimoto, K., Yamada, T.,A New Method for Management of Consumer's Private Information on Bidirectional Broadcasting Services. Japan, IEEExplore.ieee.org, 2007

[13] Olumofin, F., Goldberg, I.: Privacy Preserving Queries over Relational Databases. Technical report,
CACR 2009-37, University of Waterloo (2009)

[14] Sassaman, L., Cohen, B., Mathewson, N.: The Pynchon Gate: A Secure Method of Pseudonymous Mail Retrieval. In: ACM WPES, pp. 1-9 (2005)

[15] Sion, R., Carbunar, B.: On the computational practicality of private information

[16] Sil Choi, M., Soo Park, Y., Seon Ahn, K.: Web Site Management System through Private Information Extraction. Department of Computer Engineering, Kyungpook National University, Daegu, South Korea, IEEExplore.ieee.org (2002)

[17] Wang, S., Agrawal, D., Abbadi, A. El: Generalizing PIR for Practical Private Retrieval of Public Data. Data and Applications Security XXIV LNCS 6166, pp. 1-16, 2010

[18] Yang, E., Xu, J., Bennett, K.: Private Information Retrieval in the Presence of Malicious Failures. Proceedings of the 26th Annual Internal Computer Software and Applications Conference (COMPSC'02), IEEExplore.ieee.org