A new distributed anonymization protocol to satisfy multiple data provider's privacy requirements

Salheddine kabou
EEDIS Laboratory
Djillali Liabes University of Sidi Bel Abbes
salheddine.kabou@univ-sba.dz

Sidi Mohamed Benslimane
Computer Sciences Department
Djillali Liabes University of Sidi Bel Abbes
benslimane@univ-sba.dz

Abstract –Privacy and security concerns are among the main obstacles facing the widespread adoption of this new technology. Data anonymization makes data worthless to anyone except the owner of the data. It is one of the methods for transforming the data in such a way that it prevents identification of key information from an unauthorized person. Most of the existing works use a k-anonymat model for preserving privacy for data subject that offers lower utility. Motivated by this, we develop a new distributed anonymization protocol to satisfy multiple data providers privacy requirements based on a k-concealment model that offers a higher utility.

Keywords – Privacy, Concealment, Anonymization, Cloud Computing

1. INTRODUCTION

The confidentiality of this data must be preserved before outsourcing to the commercial public cloud, i.e. any sensitive information should not be disclosed. Data anonymization is one of the privacy preserving techniques that translate the information, making the original data worthless to anybody except the owners [1]. It has been widely discussed in the literature such as *k-anonymity* [2], l-diversity [3], *k-concealment* [4]. The objective of our work is to adopt a new distributed anonymization protocol over cloud servers, which can satisfy different data provider's personal needs and maximize the utility of the anonymous data still remains a challenging problem using an alternative model. In this paper, we offer the following contributions:

1. We present an algorithm which inserts data subjects into an R*-tree for anonymization protocol using a k-concealment model.
2. As the output of the protocol, each private dataset produces a local anonymized dataset that satisfies each data provider's privacy constraints and their union forms a global virtual database that meets a global anonymization principle (k-conealment).

The rest of the paper is organized as follows. In Section II, we give we discuss related works to our research. In Section III, presents our distributed anonymization protocol, and Section IV explain a R*-tree generalization principal to achieve the protocol. Finally, we conclude our discussion in Section V.

2. RELATED WORK

2.1 Privacy preserving data publishing for single databases

In recent years, Privacy preserving data publishing for centralized databases has been studied extensively [5](.K-anonymity,l-diversity) K-concealment proposes to generalize the table records so that each one of them becomes computationally - indistinguishable from at least k−1 others. In this study, our distributed anonymization protocol is built on top if the k-concealment principles.

2.2 Privacy preserving data publishing for multiple databases

There are a number of possible solutions that can be applied for anonymizing distributed data

The *integrate-and-anonymize* [8] solution is the simplest approach that assumes an existence of third party that can be trusted by each of the data owners. Here data owners send their data to the trusted third party where data integration and anonymization are performed. This approach is not feasible for many scenarios [7]. Finding such a trusted third party is not always feasible. Compromise of the server by hackers could lead to a complete privacy loss for all the participating parties and data subjects.

The basic idea of an alternative approach *anonymize-and-integrate* is for each data provider to perform data anonymization independently. In case of horizontal data distribution [8], this tends to have a negative impact on data quality. For example, enforcing local k-anonymity can cause the data utility to suffer more than enforcing global k-anonymity. If data is distributed vertically [9], the integration of the locally anonymous datasets can result in global non-anonymous datasets.

In *the Virtual anonymization* [6] [10] data providers participate in distributed protocols to produce a virtual integrated and anonymized database. Important to note is that the anonymized data still resides at individual databases and the integration and anonymization of the data is performed through the secure distributed protocols.

In [6] the authors propose an adaptation of the Mondrian generalization-based algorithm for producing a k-anonymous release of the union of the datasets by using a set of secure multi-party computation protocols (i.e., secure computing sum, secure median protocols). The execution sequence is under the control of a single leading site and assumes that participants are fully available. Ding et al [10] presented a distributed anonymization algorithm using R-tree index structure [11]. Their algorithm uses a greedy approach to recursively insert the data objects into the quasi-identifier domain space.

The above works have used a k-anonymity model for the privacy of data subjects. They choose it because it's useful in several practical applications; also it suffers in data utility. Here, our work is aimed at outsourcing data provider's private dataset to cloud servers for data sharing. More importantly, our anonymization protocol aims to achieve anonymity for both (1) data subjects: by using a k-concealment model which ensures the same level of security and offers higher utility than others models that have used in the above works, and (2) data providers: by designing a new distributed anonymization algorithm that uses a R*-tree structure [12]. Its offers a best choosing of appropriate insertion of data objects into the quasi-identifier domain space, and it find an adequate partitioning of the quasi-identifier domain space.

3. DISTRIBUTED ANONYMIZATION PROTOCOL

3.1 Algorithm

There are a large number of algorithms proposed to achieve k-anonymity. These algorithms have been successfully used to enhance anonymity in distributed environments. Although to our best knowledge, the model k-concealment has not yet been used for the cloud environment.

```
Algorithm.1 Master
Phase 1: Reading data (slave)
Read data (B, num) from each slave node into a set r

Phase 2: Insertion on R*-tree (Generalisation)
For each rectangle B ϵ r, do
    Finds on every level of the tree, the most suitable subtree
    to accommodate the new entry (see process of R*-tree
    generalisation)
    If split is possible, do split (see process of split)
End for

Phase 3: Modify the local $k_i$-concealment databases
For each equivalence class $E_i$ of the K-concealment table, do
Get rectangles set P contained in R* of $E_i$
        For each rectangle B ϵ P, do send B and R* to slave
              node
        End for
End for
```

Figure.2 Distributed anonymization algorithm-Master node

International Conference on Advanced Aspects of Software Engineering
ICAASE, November, 2-4, 2014, Constantine, Algeria.

```
Algorithm.2 Slave node i (i>0)
Phase 1: Sending data to the master
 For each equivalence class $E_j$ of $k_i$ concealment table, do
    Send information of $E_j$ to the Master in form (B, num)
end For
Phase 2: Receive modification from the master
Read the data B and R* from the Master
For each equivalence class $E_j$ of $k_i$-concealment table, do
    if rectangle of $E_j$ equals B
        Enlarge the rectangle of $E_j$ to R*
   end if
 end For
```

Figure.2 Distributed anonymization algorithm-Slave node

We assume that a master node is selected from cloud server for the main protocol, and all the other local databases are consider as slave nodes. The protocols for the master node and other slave nodes are presented in Algorithm1 and 2. In the first, the slave node sends a data that is abstract information of each equivalence class to the Master node in form (*B, num*). Where *B* refers a d-dimensional rectangle which is the bounding box of the equivalence class's spacial QI values $[l_1,u_1]$, ...,$[l_d,u_d]$, and *num* is the total number of data subjects in the equivalence class.

In Phase 1, the master node reads data (*B*, *num*) of each equivalence class that had sent from each slave node into a set *r.* In phase 2, the algorithm inserts each rectangle *B* from *r* into an R*-tree: for the first time, it finds on every level of the tree, the most suitable subtree to accommodate the new entry (minimum overlapping), and if split is possible, do split.

In phase 3, the master node modifies all the initial local k_i-concealment databases by traversing each equivalence class E_i of the K-concealment table, to find the rectangles set *P* contained in it.

For each rectangle *B* ϵ *P* that contained in the bounding box R* of E_i, we send data *B* and R* back to a corresponding slave node. When receive the data *B* and R* from the master, the slave node finds out the equivalence class whose rectangle equals *B* and then enlarges its rectangle to R*.

3.2. Split process

The objective is to split the data as much as possible while satisfying the privacy constraints to maximize the utility of anonymized data.

The R*-tree uses the following steps to find a better split. In a first step, the split axis has to be chosen. Along each axis, the entries are first sorted by the lower value, and then sorted by the upper value of their rectangles (rectangles of equivalence classes). For each sort M-2m+2 distributions of the M+1 (*node a*) entries into two groups are determined (M is the maximum number of entries that will fit in one node and ***m*** is a minimum number of entries), where the k-th distribution (k=1, …., M-2m+2) is determined as follows: The first group contains the first (m–1) + k entries, the second group contains the remaining entries.

For each of the M–2m+2 distributions, the margin-value is determined by summarizing the margin length of the two minimum bounding boxes (rectangles of equivalence classes) of both distributions. Finally, the axis that returns the minimum margin value is selected a split axis. In the next step, an adequate partitioning of the entries along the split axis determined. The overlap-value for each of the 2(M–2m+2) distributions is considered where the overlap-value denotes the size of the area of the overlap between the two minimum bounding boxes of both partitions. Here, the secure sum protocols [13] can be used to securely compute the minimum area-value denoting the sum of the size of the areas covered by both minimum bounding boxes of both partitions [12].

4. R*-TREE GENERALIZATION

Our algorithm uses an R*-tree. It generalizes the data by inserting it into the minimum overlapping quasi-identifier domain space and when overflow occurs, the quasi-identifier domain space will be split into two parts along the best axis chosen. It recursively chooses the best branch to inset the data objects for gathering the closest data objects. When all the data tuples are inserted into the R*-tree, the generalization table was built.

A non-leaf node contains entries of the form (*cp, B*) where *cp* is the address of the child node in the R*-tree and *B* is the minimum bounding rectangle of all rectangles which are entries in that child node. A leaf node contains entries of the form (*Oid, B*) where *Oid* refers to a record in the database, and $B = (B_1, B_2,…, B_d)$ is a d-dimensional rectangle which is the bounding box

International Conference on Advanced Aspects of Software Engineering
ICAASE, November, 2-4, 2014, Constantine, Algeria.

of one equivalence class's spacial QI values. at this point, *d* is the number of dimensions and B_i is a bounded interval [x,y] describing the QI value along dimension.

The R*-tree construction is based on the insertion algorithm [12], which focuses on: (1) Choossubtree: Beginning in the root, descending to a leaf, it finds on every level the most suitable subtree to accommodate the new object. (2) The split: If Choosesubtree ends in a node filled with the maximum number of objects M, split should distribute M+ 1 rectangles into two nodes in the most appropriate manner.

We create the R*-tree by inserting every object into the indexing structure. Given a new object, the authors in [13] consider only the area. Our algorithm tests the parameters area and overlap in different combinations. The overlap of an entry is defined as follows. Let $E_1,\dots,E_p$ be the entries in the current node. Then,

$$\text{overlap}(E_k) = \sum_{i=1, i \neq k}^{p} \text{area}(E_i \cap E_k),\ 1 < k < p$$

For choosing the best non-leaf node, we can use the method used in R-tree [11]. For the leaf nodes, minimizing the overlap performed slightly better.

5. CONCLUSION

We have presented a distributed anonymization protocol for privacy-preserving data publishing from multiple data providers in a cloud environment. Our work addresses two important issues, privacy of data subjects and privacy of data providers. For the privacy of data subjects (individuals), we have used a k-concealment model that offers a higher utility with less generalisation than that which is required by k-anonymity used in [6] and [10]. For the privacy of data providers, we have adopted a bottom-up algorithm instead of top-down approach used in [6] while using R*-tree index for better generalization. We also illustrated that the R*-tree strategy leads to a more effective insertions and splitting than that of the R-tree. We are now working on implementing this algorithm to validate our contribution.

6. REFERENCE

[1] Sedayao, "Enhancing cloud security using Data Anonymization", Intel white paper, June 2012.

[2] P. Samarati and L. Sweeney. "Generalizing data to provide anonymity when disclosing information". In ACM-SIGMOD Symposium on Principles of Database Systems (PODS), page 188, 1998.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam." l-diversity: Privacy beyond k-anonymity". In International Conference on Data Engineering (ICDE), page 24, 2006.

[4] T.Tassa, A.Mazza, A.Gionis, " k-Concealment: An Alternative Model of k-Type Anonymity", Transactions on Data Privacy 5, pp189–222, 2012

[5] B. Fung, K. Wang, R. Chen, " Privacy-preserving data publishing", A survey of recent developments. ACM Computing Surveys, CSUR (2010)

[6] P. Jurczyk, , L. Xiong, "Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers". In: Gudes, E., Vaidya, J. (eds.) Data and Applications Security 2009. LNCS, vol. 5645, pp. 191–207. Springer, Heidelberg (2009)

[7] F. Kohlmayer, F. Prasser, C. Eckert, A. Kuhn, "A flexible approach to distributed data anonymization", In Journal of Biomedical Informatics, August 2013.

[8] N. Mohammed, B. Fung , "Centralized and distributed anonymization for high-dimensional healthcare data". ACM Trans Knowl Discovery Data 2010;4(4):1–33.

[9] W.Jiang, , C.Clifton, "A secure distributed framework for achieving k-anonymity". VLDB Journal 15(4), 316–333 (2006)

[10] X.Ding, Q.Yu, J.LI, J.Liu, H.Jin, "Distributed Anonymization for Multiple Data Providers in a Cloud System", In: W. Meng et al. (Eds.): DASFAA 2013, Part I, LNCS 7825, pp. 346–360, 2013.

[11] A .Guttman,. "R-trees: a dynamic index structure for spatial searching". In: SIGMOD (1984)

[12] N. Beckmann, H.Kriegel, R.Schneider, B. Seeger, "The R*-tree: An efficient and robust access method for points and rectan-gles". In: Proceedings of the International Conference on Manage-ment of Data (SIGMOD'90), Atlantic City, NJ (1990) pp. 322–331

[13] B¨ ottcher, S., Obermeier, S, "Secure set union and bag union computation for guar-anteeing anonymity of distrustful participants", JSW 3(1), 9–17 (2008)

International Conference on Advanced Aspects of Software Engineering
ICAASE, November, 2-4, 2014, Constantine, Algeria.