Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Confidential Query Building in Cloud using KNN-Query Processing

MuthuvelS*, Dr. T. Stephen Thangaraj Ph.D, R. Nanda Kumar M.Tech
Computer Science and Engineering, T.J. Institute of Technology,
Tamil Nadu, India

Abstract— With the wide deployment of public cloud computing infrastructures, using clouds to host data query services has become an appealing solution for the advantages on scalability and cost-saving. However, some data might be sensitive that the data owner does not want to move to the cloud unless the data confidentiality and query privacy are guaranteed. On the other hand, a secured query service should still provide efficient query processing and significantly reduce the in-house workload to fully realize the benefits of cloud computing. We propose the RASP data perturbation method to provide secure and efficient range query and kNN query services for protected data in the cloud. The RASP data perturbation method combines order preserving encryption, dimensionality expansion, random noise injection, and random projection, to provide strong resilience to attacks on the perturbed data and queries. It also preserves multidimensional ranges, which allows existing indexing techniques to be applied to speedup range query processing. The kNN-R algorithm is designed to work with the RASP range query algorithm to process the kNN queries. We have carefully analysed the attacks on data and queries under a precisely defined threat model and realistic security assumptions. Extensive experiments have been conducted to show the advantages of this approach on efficiency and security.

Keywords— Query Services In The Cloud, Privacy, Range Query, Knn Query,Fairness, Mappings, Peer Data Management Systems, Peer-to-peer Data Integration

I. INTRODUCTION

We consider applications where the original sensitive data cannot be perturbed. Perturbation is a very useful technique where the data is modified and made “less sensitive” before being handed to agents. One can add random noise to certain attributes, or one can replace exact values by ranges. However, in some cases it is important not to alter the original distributor’s data. For example, if an outsourcer is doing our payroll, he must have the exact salary and customer bank account numbers. If medical researchers will be treating patients (as opposed to simply computing statistics), they may need accurate data for the patients. Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

We study unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, in this paper we develop a model for assessing the “guilt” of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding “fake” objects to the distributed set. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty. We present a model for calculating “guilt” probabilities in cases of data leakage. Then, in the second part, we present strategies for data allocation agents. Finally, we evaluate the strategies in different data leakage scenarios, and check whether they indeed help us to identify a leaker.

Data leakage can be elaborated as in when a data distributor has given sensitive data to a set of supposedly trusted agents and some of the data is leaked and found in an unauthorized place. An enterprise data leak is a scary proposition. Security practitioners have always had to deal with data leakage issues that arise from various ways like email, 1M and other Internet channels. In case of data leakage from trusted agents, the distributor must assess the likelihood that the leaked data came from one or more agents.

This can be done by using a system which can identify those parties who are guilty for such leakage even when data is altered. For this the system can use data allocation strategies or can also inject "realistic but fake" data records to improve identification of leakage. Moreover, data can also be leaked from within an organization through e-mails. Hence, there is also a need to filter these e-mails. This can be done by blocking e-mails which contains images, videos or sensitive data for an organization. Principle used in e- mail filtering is we classify e-mail based the fingerprints of message bodies, the white and black lists of email addresses and the words specific to spam.

II. PEER DATA MANAGEMENT SYSTEMS

A. Peers

In a PDMS, each peer can play any combination of three different roles: client, server, and translator. A peer is considered a client whenever it issues queries. It is considered a server whenever it processes queries and returns results from its local database. Finally, it is considered a translator whenever it uses the mappings it maintains with other peers to translate queries and results to and from these peers. Hence, in the context of processing a particular query, one peer will be acting as a client, one or more peers will be acting as servers, and zero or more peers will be acting as translators.

B. Mappings

The mappings used in a PDMS have two main purposes: query translation and result translation. For query translation, different types of mappings were proposed. For example, the Hyperion system is based on the notion of mapping tables, which encompass attribute-to-attribute and value-to-value mappings. Other PDMSs, such as Piazza , use more complex schema mappings. In particular, Piazza's mapping language supports both Local-As-View (LAV) and Global-As-View (GAV) mapping types. However, unlike Hyperion, it does not support value-to-value mappings. For the purpose of result translation, where the values in a query result need to be translated from the vocabulary of one peer to another, the Hyperion-style value-to-value mapping tables are required. Result translation is of particular importance, especially when the final result is to be machine-processed; i.e., a program has to match the values in the result with the local vocabulary (e.g. to store the result locally, and insert the correct foreign keys with each incoming record). Since the key goal of our work is to protect the privacy of the query results, we will focus on this type of mappings (i.e. mapping tables). In fact, our proposed protocol is agnostic to the type of schema mappings used.

C. P2P Network

Peers having direct mappings between them are usually referred to as acquainted peers. In most PDMSs, such as Hyperion and Piazza, the complete acquaintance graph is accessible to all peers. It can be stored in a centralized location, or replicated at each peer. The acquaintance graph only gives information about which peers are acquainted, but not the details of the mappings between each pair of acquainted peers. The latter type of information can only be obtained by sending requests to the acquainted peers, where the mappings are actually stored. Weights can be assigned to the edges of the graph to reflect one or several factors about the acquaintance link such as reliability, latency, usage cost, or a combination of these factors. The mapping tables are typically replicated in both acquainted peers. This redundancy allows any two peers having a common acquaintance (common neighboring peer) to communicate directly using the schema and vocabulary of their common acquaintance. Consequently, the shortest path between any two remote peers can be reduced to about half its original length, since every other peer can be skipped. It follows that about half the communication and processing costs are saved.

D. Query Answering

There are two types of queries that can be answered in a PDMS: broadcast queries and targeted queries. For broadcast queries, the client sends the query to its acquaintances which in turn forward it to their own acquaintances and so on, thus propagating the query to the whole network. Before a peer forwards the query, it is translated first to conform to the schema of the following peer. Each peer receiving the query and having a local database executes it locally. Finally, all the peers that could find relevant local results send those results back to the client; each through the same path that was used to deliver the query to it, but in the reverse direction. Results get translated by the intermediate peers on their way back to the client. For targeted queries, the client specifies a target peer to request the results from. The shortest path to the target peer is first computed on the acquaintance graph. The query is then sent (and translated) along the discovered path, and the corresponding result is also sent (and translated) along the same path, but in the reverse direction. For the rest of the discussion, we will focus on targeted queries rather than broadcast queries. A broadcast query can be viewed as a collection of targeted queries in terms of delivering results from different servers back to the client. Each result will travel back through a certain path, similar to targeted queries.

E. Implementation

We used the implementation of the Hyperion system as the platform to base our protocol upon. We have added support for targeted queries as well as result translation on the path back to the client. We also introduced the optimization of skipping every other intermediate peer on the path from a client to a remote target peer.

III. RASP: RANDOM SPACE PERTURBATION

In this section, we present the basic definition of Random Space Perturbation (RASP) method and its properties. We will also discuss the attacks on RASP perturbed data, based on the threat model.

RASP is one type of multiplicative perturbation, with a novel combination of OPE, dimension expansion, random noise injection, and random projection. Let's consider the multidimensional data are numeric and in multidimensional vector space1. The database has k searchable dimensions and n records, which makes a d × n matrix X. The searchable dimensions can be used in queries and thus should be indexed. Let x represent a d-dimensional record, x Rd. Note that in the d-dimensional vector space Rd, the range query conditions are represented as half-space functions and a range query is translated to finding the point set in corresponding polyhedron area described by the half spaces. The RASP perturbation involves three steps. Its security is based on the existence of random invertible real-value matrix generator and random real value generator. For each k-dimensional input vector x.

1) An order preserving encryption (OPE) scheme, Eope with keys Kope, is applied to each dimension of x: Eope(x, Kope) Rd to change the dimensional distributions to normal distributions with each dimension's value order still preserved.

2) The vector is then extended to d+2 dimensions as G(x) = ((Eopt(x))T , 1, v)T , where the (d + 1)- th dimension is always a 1 and the (d + 2)- th dimension, v, is drawn from a random real number generator RNG that generates random values from a tailored normal distributions. We will discuss the design of RNG and OPE later.

3)The (d + 2)-dimensional vector is finally transformed to F(x,K = {A,Kope,RG}) = A((Eope(x))T , 1, v)T , where A is a (d+2)×(d+2) randomly generated invertible matrix with aij R such that there are at least two non-zero values in each row of A and the last column of A is also non-zero2. Kope and A are shared by all vectors in the database, but v is randomly generated for each individual vector. Since the RASP-perturbed data records are only used for indexing and helping query processing, there is no need to recover the perturbed data. As we mentioned, in the case that original records are needed, the encrypted records associated with the RASP-perturbed records will be returned.

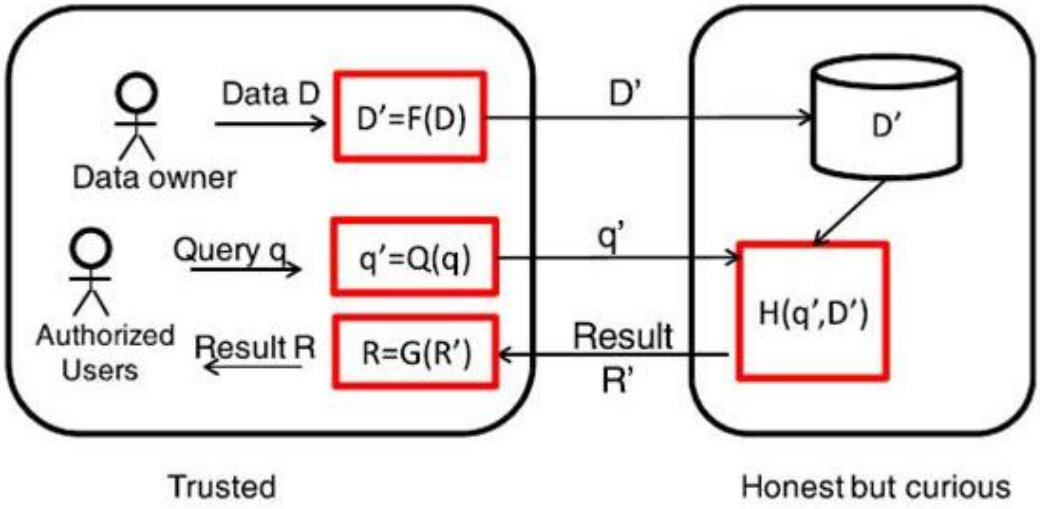

Fig.1 System Architecture for Rasp-Based Query Services.

A. Properties of RASP

RASP has several important features. First, RASP does not preserve the order of dimensional values because of the matrix multiplication component, which distinguishes itself from order preserving encryption (OPE) schemes, and thus does not suffer from the distribution-based attack. An OPE scheme maps a set of single-dimensional values to another, while keeping the value order unchanged. Since the RASP perturbation can be treated as a combined transformation F(G(Eope(x))), it is sufficient to show that F(y) = Ay does not preserve the order of dimensional values, where y ∈Rd+2andA ∈R(d+2)×(d+2).The proof is straightforward as shown in Appendix. Second, RASP does not preserve the distances between records, which prevents the perturbed data from distance-based attacks. Because none of the transformations in the RASP: Eope, G, and F preserves distances, apparently the RASP perturbation will not preserve distances. Similarly, RASP does not preserve other more sophisticated structures such as covariance matrix and principal components. Third, the original range queries can be transformed to the RASP perturbed data space, which is the basis of our query processing strategy. A range query describes a hyper-cubic area (with possibly open bounds) in the multidimensional space. Thus, we can search the points in the polyhedron to get the query results.

IV. KNN QUERY PROCESSING WITH RASP

RASP perturbation does not preserve distances (and distance orders), kNN query cannot be directly processed with the RASP perturbed data. In this section, we design a kNN query processing algorithm based on range queries (the kNN-R algorithm). As a result, the use of index in range query processing also enables fast processing of kNN queries.

The original distance-based kNN query processing finds the nearest k points in the spherical range that is centered at the query point. The basic idea of our algorithm is to use square ranges, instead of spherical ranges, to find the approximate kNN results, so that the RASP range query service can be used. There are a number of key problems to make this work securely and efficiently.

(1) How to efficiently find the minimum square range that surely contains the k results, without many interactions between the cloud and the client

(2) Will this solution preserve data confidentiality and query privacy

(3) Will the proxy server's workload increase to what extent?

The algorithm is based on square ranges to approximately find the kNN candidates for a query point, which are defined as follows. A square range is a hyper-cube that is centered at the query point and with equal-length edges. Figure 2 illustrates the range-query-based kNN processing with two-dimensional data. The Inner Range is the square range that contains at least k points, and the Outer Range encloses the spherical range that encloses the inner range. The outer range surely contains the kNN results but it may also contain irrelevant points that need to be filtered out.

The kNN-R algorithm returns results with 100% recall. Proof: The sphere in Figure 2 between the outer range and the inner range covers all points with distances less than the radius r. Because the inner range contains at least k points, there are at least k nearest neighbors to the query points with distances less than the radius r. Therefore, the k nearest neighbors must be in the outer range. The kNN-R algorithm consists of two rounds of interactions between the client and the server. Figure 4 demonstrates the procedure.

(1) The client will send the initial upper-bound range, which contains more than k points, and the initial lower-bound range, which contains less than k points, to the server. The server finds the inner range and returns to the client.

(2) The client calculates the outer range based on the inner range and sends it back to the server. The server finds the records in the outer range and sends them to the client.

(3) The client decrypts the records and find the top k candidates as the final result. If the points are approximately uniformly distributed, we can estimate the precision of the returned result. With the uniform assumption, the number of points in an area is proportional to the size of the area. If the inner range contains m points, m >= k, the outer range contains q points, and the dimensionality is d, we can derive q = 2d/2m. Thus, the precision is k/q = k/ (2d/2m). If m ˜ k and d = 2, the precision is around 0.5. When d increases, the precision decreases exponentially due to the curse of dimensionality, which suggests kNN-R should not work effectively on high-dimensional data.

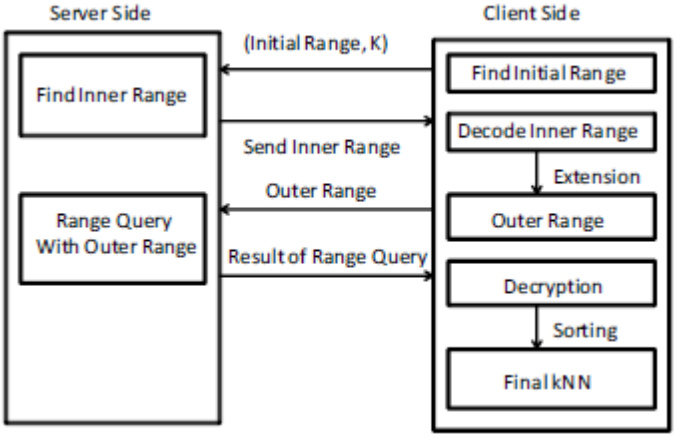

Fig. 2.Procedure of KNN-R Algorithm

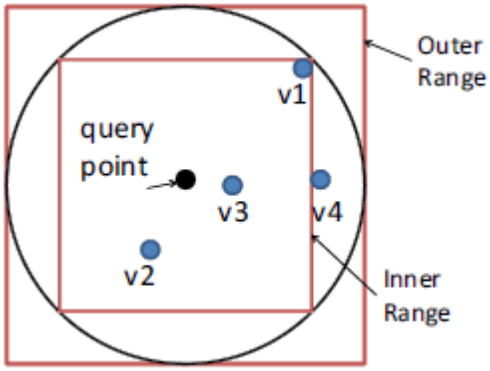

Fig. 3.Illustration of KNN-R Algorithm, If K=3

A. Finding Compact Inner Square Range

An important step in the kNN-R algorithm is to find the compact inner square range to achieve high precision. In the following, we give the (k, δ) range for efficiently finding the compact inner range.

A (k, δ)-*range* is any square range centered at the query point, the number of points in which is in the range [k, k + δ], δ is a nonnegative integer. We design an algorithm similar to binary search to efficiently find the (k, δ)-range. Suppose a square range centered at the query point with length of L in each dimension is represented as S (L). Let the number of points included by this range is N (L). If a square range S (in) is enclosed by another square range S (out), we say S (in) ⊂S (out). It directly follows that N (in) ≤ N (out), and also *Corollary 1:* If N (1) <N (2), S (1) ⊂S (2). Using this definition and notation, we can always construct a series of enclosed square ranges centered on the query point: S (L1) ⊂S (L2) ⊂. . . ,⊂S (Lm). Correspondingly, the numbers of points enclosed by {S(Li)} have the ordering N(L1) ≤ N(L2) ≤ . . .N(Lm).Assume that S(L1) is the initial range containing less than k points and S(Lm) is the initial upper bound range; both are sent by the client. The problem of finding the compact inner range S can be mapped to a binary search over the sequence {S (Li)}. In each step of the binary search, we start with a lower bound range, denoted as S (low) and a higher bound range, S (high). We want the corresponding numbers of enclosed points to satisfy N (low) < k ≤ N (high) in each step, which is achieved with the following procedure. First, we find the middle square range S (mid), where mid = (low + high)/2. If S (mid) covers no less than k points, the higher bound: S (high) is updated to S (mid); otherwise, the lower bound: S (low) is updated to S (mid). At the beginning step S (low) is set to S (L1) and S (high) is S (Lm). This process repeats until N (mid) < k + δ or high − low < E,where E is some small positive number.

B. Defining Initial Bounds

The complexity of the (k, δ)-range algorithm is determined by the initial bounds provided by the client. Thus, it is important to provide compact ones to help the server process queries more efficiently. The initial lower bound is defined as the query point. For q (q1. . . qd), the dimensional bounds are simply qj ≤ Xj ≤ qj .The higher bounds can be defined in multiple ways. Applications often have a user specified interest bound, for example, returning the nearest gas station in 5 miles, which can be used to define the higher bound. We can also use center-distance based bound setting. Let the query point has a distance γ to the distribution center - as we always work on normalized distributions, the center is (0, 0). The upper bound is defined as qj − ϙγ ≤ Xj ≤ qj + ϙγ, where epsilon ∈ (0, 1] defines the level of conservativity.If it is really expected to include all candidate kNN regardless how distant they are, we can include a rough density-map for quickly identifying the appropriate higher bound. However, this method works best for low dimensional data as the number of bins exponentially increases with the number of dimensions. In experiments, we simply use the method and 5% of the domain length for the extension.

C. Query Analysis

Private information retrieval (PIR) tries to fully preserve the privacy of access pattern, while the data may not be encrypted. PIR schemes are normally very costly. Focusing on the efficiency side of PIR, Williams et al. use a pyramid hash index to implement efficient privacy preserving data-block operations based on the idea of Oblivious RAM. It is different from our setting of high throughput range query processing. Hu et al. addresses the query privacy problem and requires the authorized query users, the data owner, and the cloud to collaboratively process kNN queries. However, most computing tasks are done in the user's local system with heavy interactions with the cloud server. The cloud server only aids query processing, which does not meet the principle of moving computing to the cloud.

D. Query Authentication

In Server Section authorized clients are added with respective MAC & IP address. In this section server maintains the log of all query processed. RASP Implementation will be controlled in this section. Detected Clone Node logs are maintained in this section. It is done by the administrator. Here every client will give their IP address and MAC address for registration. Authenticated client only can able to transfer the data. Every authorized client should have an individual IP address and MAC address. This specific address is used to identify the clone node detection.

V. DISCUSSION ANDCONCLUSION

In this paper, we have proposed an approach that identifies which part of intermediate data sets needs to be encrypted while the rest does not, in order to save the privacy preserving cost. A tree structure has been modelled from the generation relationships of intermediate data sets to analyse Privacy propagation among data sets. We have modelled the problem of saving privacy-preserving cost as a constrained optimization problem which is addressed by decomposing the privacy leakage constraints. A practical heuristic algorithm has been designed accordingly. Evaluation results on real-world data sets and larger extensive data sets have demonstrated the cost of preserving privacy in cloud can be reduced significantly with our approach over existing ones where all data sets are encrypted. In accordance with various data and computation intensive applications on cloud, intermediate data set management is becoming an important research area. Privacy preserving for intermediate data sets is one of important yet challenging research issues, and needs intensive investigation. With the contributions of this paper, we are planning to further investigate privacy aware efficient scheduling of intermediate data sets in cloud by taking privacy preserving as a metric together with other metrics such as storage and computation. Optimized balanced scheduling strategies are expected to be developed toward overall highly efficient privacy aware data set scheduling.

REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proceedings of ACM SIGMOD Conference, 2004.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. K. and Andy Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," Technical Report, University of Berkerley, 2009.

[3] J. Bau and J. C. Mitchell, "Security modeling and analysis,"

[4] IEEE Security and Privacy, vol. 9, no. 3, pp. 18–25, 2011.

[5] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge

[6] University Press, 2004.

[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy preserving multi-keyword ranked search over encrypted cloud data," in INFOCOMM, 2011.

[8] K. Chen, R. Kavuluru, and S. Guo, "Rasp: Efficient multidimensional range query on attack-resilient encrypted databases," in ACM Conference on Data and Application Security and Privacy, 2011, pp. 249–260.

[9] K. Chen and L. Liu, "Geometric data perturbation for outsourced data mining," Knowledge and Information Systems, 2011.

[10] K. Chen, L. Liu, and G. Sun, "Towards attack-resilient geometric data perturbation," in SIAM Data Mining Conference, 2007.

[11] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," ACM Computer Survey, vol. 45, no. 6,pp. 965–981, 1998.

[12] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proceedings of the 13th ACM conference on Computer and communications security. New York, NY, USA:ACM, 2006, pp. 79–88.

[13] N. R. Draper and H. Smith, Applied Regression Analysis. Wiley,

[14] 1998.

[15] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra, "Executing sql over encrypted data in the database-service-provider model," in Proceedings of ACM SIGMOD Conference, 2002.

[16] T. Hastie, R. Tibshirani, and J. Friedman, the Elements of

[17] Statistical Learning. Springer-Verlag, 2001.

[14] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy-preserving index for range queries," in Proceedings of Very Large Databases Conference (VLDB), 2004.

