RASP-QS: Efficient and Confidential Query Services in the Cloud

Zohreh Alavi, Lu Zhou, James Powers, Keke Chen
Data Intensive Analysis and Computing (DIAC) Lab, Kno.e.sis Center
Department of Computer Science and Engineering
Wright State University, Dayton, Ohio 45435, USA
{alavi.3, zhou.34, powers.4, keke.chen}keke.chen@wright.edu

ABSTRACT

Hosting data query services in public clouds is an attractive solution for its great scalability and significant cost savings. However, data owners also have concerns on data privacy due to the lost control of the infrastructure. This demonstration shows a prototype for efficient and confidential range/kNN query services built on top of the random space perturbation (RASP) method. The RASP approach provides a privacy guarantee practical to the setting of cloud-based computing, while enabling much faster query processing compared to the encryption-based approach. This demonstration will allow users to more intuitively understand the technical merits of the RASP approach via interactive exploration of the visual interface.

1. INTRODUCTION

With the wide deployment of cloud infrastructures, it has become popular to host services and big data in public clouds. This new paradigm is especially attractive for data-intensive query and analysis services for its great scalability and significant cost savings. It is well known that maintaining and mining data incurs much higher cost than initial data acquisition. By moving data services to the cloud, data owners can cut costs in almost every aspect of managing and mining data. However, data privacy is still haunting data owners' minds as the underlying infrastructure is out of their control. In particular, data owners may not be aware of information leakage, which can happen in all kinds of possibilities, if the cloud provider does not want to report the leakage.

A straightforward method is to encrypt datasets before exporting them to the cloud. However, searchable encryption is very challenging, showing limited successes in some specific areas such as document search [4]. Boneh et al. [2] showed that it is possible to construct a public-key system for range query, which is one of the basic database queries (another popular one is k nearest neighbor (kNN) query as we will discuss). However, it requires a significant amount of storage and computational costs, only applicable to linear scan of the entire database. Database queries such as range and kNN queries normally demand fast processing time (logarithmic or sublinear time complexity) with the support of indexing structures. However, if not impossible, there is no efficient indexing structure developed for encrypted data yet, which renders the current encryption schemes [2] unusable for search in large databases.

We recently proposed the RAndom Space Perturbation (RASP) method [5] for the protection of tabular data, which is secure under the assumption of limited adversarial knowledge - only the perturbed data and the data distributions are known by adversaries. This assumption is appropriate in the context of cloud computing. The RASP perturbation is a unique combination of Order Preserving Encryption (OPE) [1], dimensionality expansion, noise injection, and random projection, which provides sufficient protection for the privacy of query services in the cloud. It has a number of unique features, such as preserving the topology of range query, non-deterministic results for duplicate records, and resilience to distributional attacks [5].

We develop the *secure half-space query transformation method* that casts any enclosed range in the original space to an irregularly shaped range in the perturbed space. Therefore, we are able to use a *two-stage range query processing method*: an existing multidimensional index, such as R*-Tree in the perturbed space is used to find out the records in the bounding box of the irregularly shaped range, which is then filtered with the transformed query condition. This processing strategy is fast and secure under the security assumption.

To allow the readers to fully appreciate the intuition and the ideas behind the RASP based perturbation and query processing, we propose this RASP Query Services (RASP-QS) demonstration system. This system consists of the following major components: (1) the user interface for perturbation parameter generation that allows users to observe the details of RASP perturbation, (2) the visualization of the two-stage range query processing procedure to understand the transformed query ranges and the query results, (3) the visualization of the progressive steps in the kNN query processing that is based on RASP range query processing, and (4) the performance comparison on index-aided processing on non-encrypted data, linear-scan query processing on encrypted data [2], and the RASP query processing.

2. RASP-QS ARCHITECTURE

We assume that a cloud computing infrastructure, such as Amazon EC2, is used to host the query services and large datasets. The purpose of this architecture is to extend the

Articles from this volume were invited to present their results at the 40th International Conference on Very Large Data Bases, September 1st - 5th 2014, Hangzhou, China.
Proceedings of the VLDB Endowment, Vol. 7, No. 13
Copyright 2014 VLDB Endowment 2150-8097/14/08.

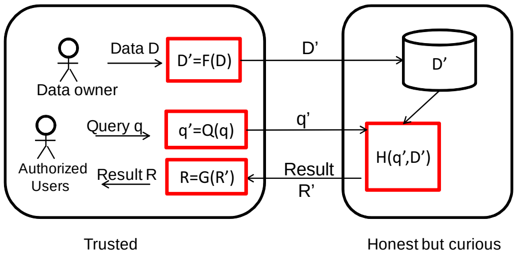


Figure 1: The RASP-QS system architecture.

proprietary database servers to the public cloud, or use a hybrid private-public cloud to achieve scalability and reduce costs while still maintaining confidentiality.

Each record x in the outsourced database contains two parts: the RASP-processed attributes $D' = F(D, K)$ for indexing and query processing, and the encrypted original records, $Z = E(D, K')$, for lossless record retrieval, where K and K' are keys for perturbation and encryption, respectively. Figure 1 shows the system architecture for both RASP-based range query service and kNN query service. There are two clearly separated groups: the trusted parties and the untrusted parties. The trusted parties include the data/service owner, the in-house proxy server, and the authorized users who can only submit queries. The data owner exports the perturbed data to the cloud. The authorized users can submit range queries or kNN queries to learn statistics or find some records. The untrusted parties include the curious cloud provider who hosts the query services and the protected database. The RASP-perturbed data will be used to build indices to support query processing.

There are a number of basic procedures in this framework: (1) $F(D)$ is the RASP perturbation that transforms the original data D to the perturbed data D'; (2) $Q(q)$ transforms the original query q to the protected form q' that can be processed on the perturbed data; (3) $H(q', D')$ is the query processing algorithm that works on D' and q' and returns the result R'. It is also possible to securely compute some statistics such as SUM or AVG of a specific dimension, if the original data records $Z = E(D, K')$ are encrypted by dimension and with a partial homomorphic encryption such as Paillier encryption. The result is recovered with the procedure $G(R')$.

2.1 RASP Perturbation

RASP perturbation is a novel combination of order preserving encryption (OPE) [1], dimension expansion, random noise injection, and random projection. Let's consider the multidimensional data are numeric and in multidimensional vector space. The database has d *searchable* dimensions, which can be used in queries, and n records, which makes a $d \times n$ matrix X. Let x represent a d-dimensional record, $x \in \mathbb{R}^d$. Note that in the d-dimensional vector space $\mathbb{R}^d$, a range query is represented as an intersection of half-space functions and a range query is translated to finding the point set in corresponding polyhedron area described by the half spaces. In a normal setting, the searchable dimensions will be indexed with techniques such as R-Tree for fast query processing.

The RASP perturbation involves three steps. For each d-dimensional input vector x,

1. An OPE scheme, E_{ope} with keys K_{ope}, is applied to each dimension of x: $E_{ope}(x, K_{ope}) \in \mathbb{R}^d$ 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 $G(x) = ((E_{opt}(x))^T, 1, v)^T$, where the $(d+1)$-th dimension is always a 1 and the $(d + 2)$-th dimension, v, is drawn from the standard normal distribution $N(0, 1)$, with the condition $v >= v_0$ (e.g., $v_0 = -3$ which covers more than 99% of the population).

3. The $(d + 2)$-dimensional vector is finally transformed to

$$F(\mathbf{x}, K = \{A, K_{ope}\}) = A((E_{ope}(x))^T, 1, v)^T, \qquad (1)$$

where A is a $(d + 2) \times (d + 2)$ randomly generated invertible matrix with $a_{ij} \in \mathbb{R}$ such that there are at least two non-zero values in each row of A and the last column of A is also non-zero.

K_{ope} and A are shared by all vectors in the database, but v is randomly generated for each individual vector, which makes the transformation non-deterministic. Since the RASP perturbed data records are only used for indexing and helping query processing, there is no need to recover the perturbed data. In the case that original records are needed, the encrypted records associated with the RASP-perturbed records will be returned. In our journal paper [5], we have proven that this perturbation method is secure against adversaries who know only the dimensional distributions and the perturbed data.

2.2 Query Transformation

A range query condition, say $X_i < a_i$, is first transformed to OPE transformed domain, i.e., $E_{ope}(X_i) \le E_{ope}(a_i)$. According to the design of the extended $(d+2)$-th noise dimension v in the RASP perturbation, v is always greater than v_0. Thus, the condition $E_{ope}(X_i) \le E_{ope}(a_i)$ is equivalent to $(E_{ope}(X_i) - E_{ope}(a_i))(v - v_0) \le 0$. Using vectors to represent the half-space conditions and u to represent the perturbed vector, we get $E_{ope}(X_i) - E_{ope}(a_i) = w^T A^{-1} u$ where $w_i = 1, w_{d+1} = -E_{ope}(a_i)$, and $w_j = 0$ for $j \ne i, d + 1$; and similarly, $v - v_0 = \mathbf{q}^T A^{-1} u$, where $q_{d+2} = 1$, $q_{d+1} = -v_0$, and $q_j = 0$, for $1 \le j \le d$. Thus, we get the transformed quadratic query condition

$$u^T (A^{-1})^T w q^T A^{-1} u \le 0. \qquad (2)$$

Let $\Theta_i = (A^{-1})^T w q^T A^{-1}$. Now the server can use $u^T \Theta_i u \le 0$ to filter out the results. In paper [5], we have proven that the query transformation is also secure against adversaries who know only the dimensional distributions and the perturbed data. However, it apparently does not preserve the privacy of access pattern, which is less important under the cloud-based assumption.

2.3 Two-Stage Fast Range Query Processing

Because the OPE transformation is typically non-linear, an enclosed range defined by half-space conditions is transformed to *a nonlinear manifold* with some unknown shape. However, we have proven that the shape is convex, which allows us to efficiently find its bounding box [5]. Therefore, we use the following two-stage processing strategy to efficiently find the query results.

Specifically, the proxy in the client side finds the maximum bounding box (MBR) of the shape (as a part of the submitted transformed query), and then submits the MBR and a set of transformed query conditions $\{\Theta_1, \ldots, \Theta_m\}$ to the server. The server uses the multidimensional tree index to find the set of records enclosed by the MBR, which are then filtered by the conditions $u^T heta_i u < 0$. The result is the *exact* result of the range query, which significantly reduces the post-processing cost that the proxy server needs to take. It is very important if the client is light-weighted such as mobile phones.

2.4 KNN-R Query Processing

The kNN-R algorithm uses *square ranges* around the query point to find the candidate nearest records. In Figure 2, the *inner square range* starts from the query point and expands until k points are included. The exact kNN result should be in the bounding sphere of the inner range, which in turn is approximated by the bounding box of the sphere. Figure 2 shows the scenario of finding the candidate set for a 3-NN query based on square ranges.

The inner range expansion can be achieved by a *binary range search* algorithm. The user can set the initial *outer square range* with a certain distance from the query point. In each iteration, the algorithm finds the middle range between the inner range and the outer range, in which if the number of enclosed points is larger than k, the outer range is replaced by the middle range; otherwise, the inner range is replaced by the outer range. This iterative process can exponentially reduce the search range and find the result quickly. The records in the final range is sent back to the client for final kNN filtering. Note that this process utilizes the linear property of the transformed queries to derive the queries for the middle range, which does not require the client's participation [5]. Experiments show that this algorithm is very efficient.

3. DEMONSTRATION

Introduction. The first part uses a poster and slides to outline the unique features of the RASP query processing approach, and introduces the viewer to the RASP-QS demonstration system.

Live System. Next, a fully interactive demonstration of the RASP-QS system will be presented. The user will be able to use the client-side system to prepare perturbed data, use the visual range/kNN query composition system to create and submit queries to the server, and observe how the query is processed by the two-stage algorithm for a range query or the kNN-R algorithm for a kNN query. The visualization of query processing step will be used to help users understand the query processing steps.

Comparison with Existing Approaches. We will also compare the performance with other related approaches. This helps users understand the unique advantages and possible limitations of the proposed approach.

3.1 Demonstration Workflow

The demonstration uses the following workflow for range query.

1. The user can choose one of the sample datasets for generating the corresponding perturbation parameters, i.e., the OPE mapping function and the invertible matrix A. The dataset is perturbed locally in the client program and then sent to the server. Each perturbed record is also associated with the encrypted original record that is encrypted with a standard encryption algorithm. Once the server has received the exported data, it builds a multidimensional R*-Tree index on the perturbed records.

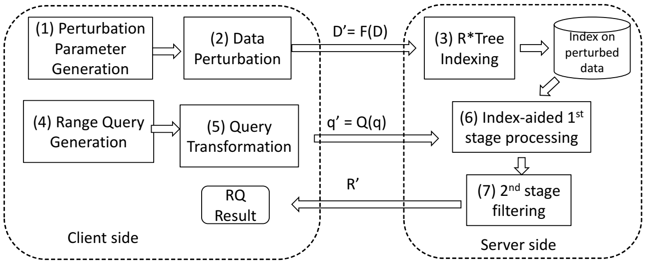


Figure 4: Major components in the demo system (for range query)

2. The user formulates a range query, which can be done visually for two-dimensional data, or by manually typing in the range definition for higher dimensional data. The transformed query matrices Q_i will be sent to the server. Users can also visually check these query matrices.

3. On receiving the query matrices, the server will apply the two-stage query processing, the whole process of which can also be visualized on the client side, so that the user can understand how the two-stage algorithm works. The query result (the encrypted original records) is sent to the client side.

Figure 4 shows the range-query workflow and the major components in the system. The kNN query processing follows a similar workflow, while including additional interactions between the client and the server to derive the final compact range after the iterative inner range expansion algorithm finishes.

3.2 Live System

We introduce the major components of the demonstration system: data perturbation, query transformation, RASP range query processing and visualization, and kNN-R query processing and visualization. The visualization part is supported by the VISTA multidimensional visualization system developed by us several years ago [3]. The server processing components will be implemented with C++ and work as web services, while the client interface is implemented with Java GUI and/or web pages.

Data Perturbation. We will prepare a set of sample datasets for the demonstration, which includes at least one two-dimensional dataset and a few higher dimensional datasets. The purpose of low dimensional data is for easier visualization and visual validation. The data perturbation component allows the user to select one of the sample datasets. The perturbation parameters have to be generated according to the specific dataset, because the OPE parameters are dataset-specific and the size of matrix A is subject to the dimensionality of the dataset. The perturbed data is sent to the server. The server then conducts multidimensional indexing on the perturbed data space.

Range Query Transformation and Processing. The query transformation and processing method is the key of RASP query processing. We develop an interactive visual

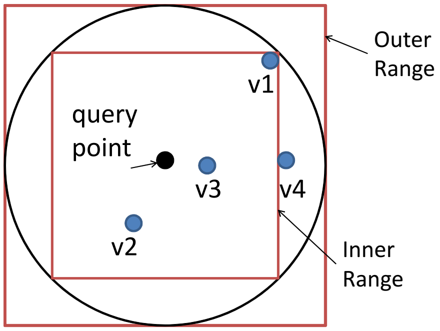
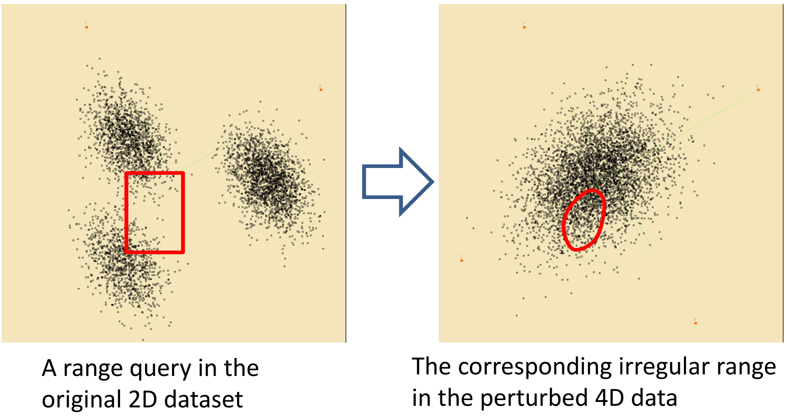


Figure 2: Illustration of kNN-R Algorithm for k=3


A range query in the original 2D dataset

The corresponding irregular range in the perturbed 4D data

Figure 3: Conceptual design for the visualization of query transformation and query result in the demonstration system.

interface to help users understand and appreciate the ideas behind the method. The query transformation method maps a linear half-spaces based range query to a quadratic surface query in the perturbed space. It is easier to understand this transformation by visualization. Using the simplest two-dimensional case for example, we can directly visualize the exact distribution of the original data. However, the perturbed data is in four-dimensional space, which has to depend on the multidimensional data visualization system VISTA [3]. Figure 3 illustrates a rectangle range on the original dataset and its hypothetical corresponding irregular range in the perturbed data space. In the final demonstration system, we will highlight the records enclosed by the range instead. For each query, we will show the complete set of transformed conditions (i.e., the matrix Θ_i for each half-space condition), the enclosed records in the perturbed space, and the bounding box that contains the irregularly shaped range for index-aided processing. Several statistics will be shown, including the number of block accesses, the number of records in the bounding box, the number of records in the final result, and the time distribution in different steps.

kNN Query Processing. In kNN query processing, we will develop a visualization interface to show the procedure of the binary square range query, which starts with the initial inner and outer ranges and progressively extends the inner range until reaching the tight bound. In each step, we will visualize the changed inner and outer ranges in the perturbed space. Some statistics will be shown as well, such as the number of iterations, the total block accesses, the number of records returned by the server, and the total time cost.

3.3 Performance Comparison

To further understand the advantages of the RASP approach, we want to show the comparison with the R*-Tree supported query processing on the original data, and the sequential scan on the encrypted data (e.g., the work on range query [2]). These methods will also be implemented with C++ for fair performance evaluation. We will let the user generate a batch of random queries with a specific size of range for a selected dataset. All the queries will be sequentially submitted by the client side and processed by the server. We will show the average query processing time cost in the server side, the server storage cost, and the client-side pre-processing and post-processing costs. We expect that the RASP approach will have much lower storage cost, query processing time, and client-side processing costs, than other methods that depend on encryption and linear scan.

4. SUMMARY

The purpose of this demonstration is to show the key ideas of the RASP-based query processing approach for efficiently and confidentially hosting query services in public clouds. This demonstration system will be highly interactive and visual, allowing the users to easily understand the technical details and appreciate the advantages of this approach. Users of the demonstration system can manipulate the system to generate perturbation parameters, observe the key steps in query processing, and evaluate the performance of several related approaches. The technical details of the RASP approach have been published recently in the journal paper [5], for which this demonstration system will be a valuable addition. We believe that the RASP approach will be a significant step towards practical confidential query services in public clouds. This work is partially supported by NSF Award 1245847.

5. REFERENCES

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

[2] Boneh, D., and Waters, B. Conjunctive, subset, and range queries on encrypted data. In *the Theory of Cryptography Conference (TCC* (2007), Springer, pp. 535–554.

[3] Chen, K., and Liu, L. VISTA: Validating and refining clusters via visualization. *Information Visualization 3*, 4 (2004), 257–270.

[4] Curtmola, R., Garay, J., Kamara, S., and Ostrovsky, R. Searchable symmetric encryption: improved definitions and efficient constructions. In *ACM CCS* (2006), pp. 79–88.

[5] Xu, H., Guo, S., and Chen, K. Building confidential and efficient query services in the cloud with rasp data perturbation. *IEEE Transactions on Knowledge and Data Engineering 26*, 2 (2014).