kNN-R: Building Secure and Efficient Outsourced kNN Query Service with the RASP Encryption

Huiqi Xu
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Computer Sciences Commons

Repository Citation
Xu, Huiqi, "kNN-R: Building Secure and Efficient Outsourced kNN Query Service with the RASP Encryption" (2012). Browse all Theses and Dissertations. 575.
https://corescholar.libraries.wright.edu/etd_all/575

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
kNN-R: Building Secure and Efficient Outsourced kNN Query Service with the RASP encryption

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

by

Huiqi Xu
B.E., Chongqing University, 2009

2012
Wright State University
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Huiqi Xu ENTITLED kNN-R: Building Secure and Efficient Outsourced kNN Query Service with the RASP encryption BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Keke Chen, Ph.D
Thesis Director

Mateen Rizki, Ph.D
Chair, Department of Computer Science and Engineering

Committee on Final Examination

Keke Chen, Ph.D

Guozhu Dong, Ph.D

Krishnaprasad Thirunarayan, Ph.D

Andrew Hsu, Ph.D
Dean, School of Graduate Studies
With the wide deployment of public cloud computing infrastructures, outsourcing database services to the cloud has become an appealing solution to save operating expense. However, some databases might be so sensitive or precious that the data owner does not want to move to the cloud unless the security is guaranteed. On the other hand, a secure outsourced service should still provide efficient query processing and significantly reduce the inhouse workload to fully realize the benefits of outsourcing. We summarize these key features for an outsourced service as the CPEL criteria: data Confidentiality, query Privacy, Efficient query processing, and Low inhouse workload. Bearing the CPEL criteria in mind, we propose an encryption called RASP to provide query services for range query and k nearest neighbors in secure outsourced databases. In the RASP encryption, data confidentiality and query privacy are guaranteed when applying it for range query and kNN. Efficient query processing is achieved by two aspects: (1) all encrypted data can be indexed to speedup query processing using RTree; (2) The protocol for k nearest search in outsourced databases can find high precision kNN results, which also minimizes costs between the cloud server and the inhouse client. High precision kNN results and minimized interactions result in low inhouse workload. In addition, we have conducted a thorough security analysis on data confidentiality and query privacy.
## Contents

1 Introduction .................................................................................. 1
2 Random Space Encryption ............................................................... 3
  2.1 System Architecture and Threat Model ...................................... 3
  2.2 Definitions and Notations .......................................................... 5
  2.3 RASP Data Encryption ............................................................... 5
  2.4 Analysis of Attacks on RASP-Perturbed Data .............................. 9
3 Range Query Processing with RASP ................................................ 12
  3.1 Query Transformation ............................................................... 13
  3.2 Secure Query Transformation Countering Query-based Attacks ..... 14
4 KNN-R: Using Range Queries to Process kNN Queries ................... 15
  4.1 Overview of the kNN-R Algorithm ............................................. 15
  4.2 Definitions of Compact Inner Square Range .............................. 18
  4.3 Algorithms for Finding Inner Square Range .............................. 19
  4.4 Finding Inner Range with Secure Range Queries ....................... 22
  4.5 Finding Compact Initial Range with Density Map ..................... 27
  4.6 Attacks on kNN Queries ........................................................... 29
5 Experiments .................................................................................. 30
  5.1 Setup ....................................................................................... 30
  5.2 Scalability of Proxy Server ........................................................ 31
  5.3 k Minimum Range and (k, δ)-Range Algorithms .......................... 31
  5.4 Cost Reduction with Preprocessing ............................................ 33
  5.5 kNN-R vs Linear Scan Approaches ........................................... 33
  5.6 Comparing kNN-R with the Casper Approach ............................ 34
6 Related work ............................................................................... 36
7 Conclusion ................................................................................... 37
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The kNN-R System Architecture.</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Illustration of the two-stage range query processing algorithm.</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Procedure of KNN-R algorithm.</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Illustration for kNN-R Algorithm when k=3</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Calculating Points in the Circle.</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>Illustrate of the attack on the MBR of narrow ranges.</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>Performance and result precision for different $E$ setting of the $k$ minimum range algorithm.</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>Performance and result precision for different $\delta$ setting of the $(k, \delta)$-range algorithm.</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>The effect of security setting to result precision for Casper and kNN-R.</td>
<td>35</td>
</tr>
</tbody>
</table>
List of Tables

1. Comparison between Proxy and Server .................................................. 31
2. Cost reduction brought by the preprocessing algorithm. ......................... 33
3. Performance comparison on linear scan and index-aided kNN-R processing. 34
Acknowledgement

Foremost, my sincere gratitude goes to my advisor Prof. Keke Chen for accepting me to join the lab and for his kindness, motivation, expertise, and most of all, for his patience. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master study.

Besides my advisor, I would like to thank the rest of my thesis committee members: Prof. Guozhu Dong and Prof. TK Prasad, for their insightful comments, and hard questions.
Dedicated to

my dear parents, Yunfu Xu and Woxi Xiao

my fiancee, Rongyi Li

without whom it would never have been accomplished


1 Introduction

Outsourcing data-intensive services to service providers is increasingly popular with the great advantages of saving hardware and software maintenance cost. Range query and k nearest neighbors (kNN) search on large-scale databases are the important data services to many applications, from location-based services (LBS) [19], machine learning [18], to similarity search in multimedia database [25]. Once the kNN query service is outsourced, data confidentiality and query privacy become the important issues, because the data owner loses the control over the data. Adversaries, such as curious service providers, will try to breach the content of the database or intercept users’ queries to breach users’ privacy, especially when the queries are location queries [19, 9]. This security requirement dramatically increases the complexity of constructing a practical outsourced database services.

Let’s study a typical application scenario of location-based services. The LBS service provider purchased a location database to serve authorized users’ queries for either range query or k nearest neighbors search. Now the LBS service provider plans to use the cloud infrastructure, such as Amazon EC2, to reduce the cost of its service, but the service provider does not want any unauthorized party to use their location database. Considering the cloud provider may passively eavesdrop the data flow, the LBS service provider needs to come up with a solution that meets the following criteria: (1) the confidentiality of the outsourced data should be protected from attacks; (2) the privacy of users’ queries needs to be preserved; (3) queries should be efficiently processed by the cloud server to accommodate possibly large query traffic; (4) the LBS service provider’s inhouse infrastructure (it is the client to the outsourced service) should take low responsibility in helping the server process queries, so that the goal of cost reduction is achieved. We name them CPEL criteria: data Confidentiality, query Privacy, Efficient query processing, and Low inhouse workload. These criteria are also general to all kinds of outsourced services. Some related approaches have been developed to address the problem of outsourced kNN services. However, they do not achieve a good balance on these four criteria. We have summarized the weaknesses
of these approaches in Related Work (Section 6).

To achieve better CPEL criteria for the kNN service, we propose RASP encryption that builds our outsourced databases to process range queries and k nearest neighbors search. The key components include (1) the server-side efficiency and secure query processing; and (2) the data confidentiality and query privacy guaranteed by the RASP encryption. This approach has a number of unique features.

- The outsourced data is encrypted with the RASP encryption showing resilience to various types of attacks on both the outsourced data and queries.

- The query services for range query and k nearest neighbors search can be implemented with indexed outsourced data - a unique benefit of the RASP encryption. Index-aided query processing guarantees the performance.

- The kNN-R query processing also guarantees 100% recall and high precision around 50%. In addition, this high precision is not subject to the change of the user defined level of confidentiality and privacy, which is a nice property that other approaches such as Casper [19] do not offer. High precision guarantees the low workload of inhouse post-processing.

We have evaluated our approach in experiments. The results show its unique advantages on the CPEL criteria over other related approaches.

In Section 2 we introduce the system architecture of our outsourced databases, give the definition of RASP encryption and analyse the threat model for data attacks. In Section 3 we describe the algorithm of processing range query on RASP-encrypted databases. The algorithm of finding k nearest neighbors is presented in Section 4 with detailed formal analysis on its complexity and performance tradeoffs. Finally, we discuss some related approaches in Section 6.
2 Random Space Encryption

In this section, we propose the basic RAndom SPace Encryption (RASP) approach for encrypted outsourced databases. First we give the system framework and assumptions held for different threat models. Second, we define and present what the RASP encryption is. Finally, we give an analysis about how our encryption model can be resilient from data attacks.

2.1 System Architecture and Threat Model

System Architecture. Figure 1 shows how our model works to provide query services. The data owner encrypts the original data in its inhouse proxy server with the RASP encryption and uploads them to the service provider. The proxy server is the gateway for query processing. It encrypts queries, submits them to the service provider, and decrypts the query results returned from the service provider. The proxy server maintains the security keys, which are used in outsourcing data and encrypting queries. The traffic between the proxy server and the service provider contains only the encrypted data and queries. Although the proxy server does not handle the large dataset and process queries, it might still be a bottleneck for a large number of users and frequent query submissions. However, the cost to scale the proxy server should be much lower than that to host the entire query processing service and we will compare the time cost between the proxy server and the service provider.

Threat Model. We aim to protect the confidentiality of the outsourced data and the privacy of query. While query integrity is also an important issue, it is orthogonal to our study. Authentication techniques [23, 21, 13] can be integrated into our framework to address the integrity problem. Thus, the integrity problem will be excluded from the paper. We can assume the curious service provider is interested in the data and queries, but it will follow the protocol to provide the service. Also, we assume that the attacker knows the algorithms to encrypt data and queries. Active attackers will also try to obtain as much prior knowledge
as possible to break the encryption. According to the level of prior knowledge the attacker may have, we categorize the attacks into three categories.

- **Level 1**: The attacker observes only the encrypted database and encrypted queries. She/he is interested in the distributions (e.g., dimensional distributions) of the original database and the original data records. This corresponds to ciphertext-only attack (COA) in cryptography [12].

- **Level 2**: Apart from the encrypted database, the attacker knows the original data distributions and wants to recover the original records.

- **Level 3**: In addition to the above knowledge, the attacker manages to obtain a number of plaintext records/queries and their ciphertext images. This corresponds to the chosen plaintext attack (CPA) in the cryptography.

The three levels also correspond to the difficulty level of obtaining the required prior knowledge. We will analyze the security of our approach based on the three levels of attacker models.
2.2 Definitions and Notations

Let \( \mathbb{R}^d \) represent the \( d \) dimensional real domain. We define a \( d \)-dimensional point \( x_i, x_i \in \mathbb{R}^d \), as a column vector \( (x_{i1}, x_{i2}, \ldots, x_{id})^T \), \( x_{ij} \in \mathbb{R} \). The \( j \)-th dimensional value \( x_{ij} \) is treated as a sample value from the distribution of random variable \( X_j \), which distributes over a continuous real domain \( D_j \) with domain length \( |D_j| \). A database \( M \) is a set of \( d \)-dimensional points \( \{x_1, x_2, \ldots, x_n\} \), where \( n \) is the number of points.

For convenience, we use closed ranges in the following description. A \( d \) dimensional closed range \( S \) consists of the dimensional range \( [s_{i,min}, s_{i,max}] \) for the \( i \)-th dimension, where \( s_{i,min} \) and \( s_{i,max} \) define the lower and upper bounds, respectively. The length of dimension \( i \) of the range \( S \) is defined as \( s_{i,max} - s_{i,min} \). A range query is defined as a query that returns the records enclosed by the range \( S \). A kNN query is a point-based query that returns \( k \) nearest neighbors of the query point.

2.3 RASP Data Encryption

Random Space Encryption(RASP) is one type of multiplicative perturbation \cite{5}. Let \( x_i \) be a plaintext record and \( y_i \) be a ciphertext record. \( x_i \) has \( d \) dimensions and \( y_i \) has \( d+2 \) dimensions. The RASP involves two steps.

1. the vector of \( x_i \) is first extended to \( d + 2 \) dimensions as \( (x_i^T, 1, v)^T \), where the \( (d + 1) \)-th dimension is always a 1 and \( v \) is drawn from \( (0, \alpha] \) using a random number generator \( R_\alpha \), with some private \( \alpha \) and distribution.

2. After this extension the \( (d + 2) \)-dimensional vector is then subjected to the transformation

\[
y_i = A \begin{pmatrix} x_i \\ 1 \\ v \end{pmatrix},
\]

where \( A \) is a \( (k + 2) \times (k + 2) \) 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.

The proxy server keeps only the key matrix $A$ for data and query encryption. Since the noise component $v_i$ is not kept, the encrypted data $y_i$ incorporates the random noise $v_i$ that is not traceable, which increases the security. Note the $d$ dimensions attending the RASP encryption are only those dimensions in the database meaningful for query service. A database may contain more dimensions, which can be in other types, e.g. texts, or images. In practice, we use a conventional encryption to encrypt/decrypt the entire record, while only using RASP to encrypt the $d$ dimensions for indexing and querying.

RASP has two important features. First, RASP does not preserve the ordering of dimensional values, which distinguishes itself from order preserving encryption schemes, and thus does not suffer from attacks with prior knowledge. Second, RASP is convexity preserving, so a convex set $S$ such as range query can be transformed into another encrypted convex set $S'$ in the encrypted databases. Any point in $S$ cannot be mapped to a point outside $S'$ and any point not in $S$ cannot be mapped to a point in $S'$.

**RASP is Not Order Preserving.** Let $x$ be any record in the dataset, and $f^i$ be the selection vector $(0, \ldots, 1, \ldots, 0)$ i.e., only the $i$-th dimension is 1 and other dimensions are 0. Then, $(f^i)x$ will return the value at dimension $i$ of $x$.

**Proposition 1.** Suppose $A$ is the key matrix for RASP encryption, $s$ and $t$ are both extended original data points with random noises. $s'$ and $t'$ are the encrypted data points. $s_i, t_i, s'_i, t'_i$ are the values at the $i$-th dimension of each corresponding vector. Then $(s_i - t_i)(s'_i - t'_i)$ could be either positive or negative.

**Proof:** we have $s'_i = (f^i)As$ and $t'_i = (f^i)At$. Thus, we get

$$
(s_i - t_i)(s'_i - t'_i) = (s_i - t_i)(f^i)^TA(s - t)
= (s_i - t_i)\sum_{j=1}^{k} a_{i,j}(s_j - t_j),
$$

(2)
where $a_{i,j}$ is the $i$-th row $j$-th column element of $A$. Without loss of generality, let’s assume $s_i > t_i$ (for $s_i < t_i$ the same proof applies). It is straightforward to see that given the fixed values of $A$, the values of $s_j$ and $t_j$ for all $j \neq i$ can be chosen so that $(s_i - t_i) \sum_{j=1}^{k} a_{i,j} (s_j - t_j)$ is either negative or positive. Note that since each row of $A$ has at least two non-zero entries, even if $a_{i,i} (s_i - t_i)^2 > 0$ ($or < 0$), using the other non-zero value in the $i$-th row of $A$, say $a_{i,k}$, the sign of $(s_i - t_i) \sum_{j=1}^{k} a_{i,j} (s_j - t_j)$ can be adjusted to either positive or negative by appropriately choosing the values $s_k$ and $t_k$.

**RASP is Convexity Preserving.** Let’s treat data records as points in a real multidimensional space. In the following, we will show that although RASP does not preserve ordering, it preserves convexity, which forms the basis of our query processing strategy. The following definitions of convex set and convexity preserving function can be found in most textbooks on convex optimization, e.g., [3].

**Definition 1.** A set $S$ is a convex set, if and only if for $\forall x_1, x_2 \in S$, and $\forall \theta \in [0, 1]$,

$$\theta x_1 + (1 - \theta) x_2 \in S$$

**Definition 2.** A convexity preserving function $E()$ preserves the convexity of sets. Concretely, if $S$ is a convex set in the original data space, the function $E()$ always transforms $S$ to another convex set $E(S)$.

The following proposition cited from [3] is critical to the proof of convexity preserving property of RASP encryption and our query processing algorithm.

**Proposition 2.** (1) Every convex set is a (possibly infinite) intersection of halfspaces, i.e., $\bigcap H_i$, where $H_i$ defines a halfspace; and (2) the intersection of (possibly infinite) convex sets is also convex.

With this proposition we can prove that
**Proposition 3.** RASP encryption is convexity preserving.

*Beweis.* We assume an original convex set in $\mathbb{R}^k$ (that is closed) is the intersection of a set of halfspaces $\bigcap H_i$, where a halfspace $H_i$ can be represented as $w_i^T x \leq a_i$ (“=” for the closed set), and $w_i \in \mathbb{R}^k$ and $a_i \in \mathbb{R}$ are parameters for the halfspace. By replacing $x$ with a column vector $z = (x^T, 1, v)^T$ and $w_i$ with $u_i = (w_i^T, -a_i, 0)^T$, the set enclosed by $H_i$ is transformed to the set enclosed by the halfspace $H_i^{ext}: u_i^T z \leq 0$. With the RASP function, we have $y = Az$, and thus this halfspace $H_i^{ext}$ can be further transformed to $H_i'$ as follows

$$u_i^T A^{-1} y \leq 0. \quad (3)$$

Each of the halfspace conditions, $H_i'$, in the transformed space represents a convex set. Thus, the intersection of them is convex as well. Therefore, the RASP encryption is convexity preserving.

Since a range query defines a convex set, the transformation method (Eq. 3) gives a basic method for transforming the range query for the RASP encrypted data - we name it RASP query transformation method. The following proposition shows that by searching with the transformed conditions in the encrypted data space, we can get the exact set of points that is the image of the query result in the original data space.

**Proposition 4.** Let $H_i$ and $H_i'$ be halfspaces defined as in the proof of Proposition 3. The RASP query transformation uniquely maps the convex set $S$ enclosed by halfspaces $\bigcap H_i$ to the convex set $S'$ enclosed by $\bigcap H_i'$.

Based on this proposition, It is straightforward to show that by using the RASP query transformation, any point in $S$ cannot be mapped to a point outside $S'$ and any point not in $S$ cannot be mapped to a point in $S'$, which is an important character for query services.
2.4 Analysis of Attacks on RASP-Perturbed Data

As the threat model describes, attackers are interested in recovering the original data records from the perturbed data. According to the three levels of knowledge the attacker may have, we categorize the attacks on the perturbed data into three classes: (1) Naive estimation; (2) Distribution-based Attacks; and (3) Known Input/Output Attacks. Because of the random component in the RASP encryption, the best the attacker can do is to estimate the original records based on the perturbed data and the prior knowledge. If the estimation result is sufficiently accurate (above certain accuracy threshold), we say the perturbation is not secure. We will define the attacks and then analyze their effectiveness. The attacks on the queries will be discussed later in query processing.

Measuring Effectiveness of Attack

Because all attacks under the threat model are estimation attacks, we use the commonly used mean-squared-error (MSE) to evaluate the effectiveness of attack. To be semantically consistent, each dimension can be treated as sample values drawn from a random variable. Let $x_{ij}$ be the value of the $i$-th original record in $j$-th dimension and $\hat{x}_{ij}$ be the estimated value. The MSE for the $j$-th dimension can be defined as

$$MSE(X_j, \hat{X}_j) = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \hat{x}_{ij})^2.$$  

It is equivalent to the variance of $x_{ij} - \hat{x}_{ij}$. For comparing MSE for multiple columns, we also standardize these two series $\{x_{ij}\}$ and $\{\hat{x}_{ij}\}$, before calculating the MSE. The standardization procedure [8] is described as follows. Assume the mean and variance of the series $\{x_{ij}\}$ is $\mu_j$ and $\sigma_j^2$, correspondingly. The series is transformed by $x_{ij} \leftarrow (x_{ij} - \mu_j)/\sigma_j$. A similar procedure is also applied to the series $\{\hat{x}_{ij}\}$. For an attack that can only result in low-accuracy estimation, we call the RASP-perturbed dataset is resilient to that attack.

Naive Estimation. With the level 1 knowledge, the attacker observes only the perturbed data. Because none of the important relationships such as value order, distance, or statistics, is preserved, we have not discovered attacks that can obtain meaningful estimation of the original data, and believe this type of attacks is not realistic.
**Distribution-based Attack.** With the level 2 knowledge, the attacker also knows dimensional distributions. This knowledge can be possibly used to perform more effective attacks. In particular, when the original data have independent dimensions and no more than one dimension having normal distribution, an attack called Independent Component Analysis (ICA) [11] can be applied to estimate the original data from the perturbed data. Originally developed for signal processing, ICA is used to discover components $A$ (the mixing matrix) and $X$ (the original signals) from the mixed data $Y = AX$. Since ICA recovers columns in an arbitrary order, it has to rely on the known distributional information to distinguish the columns and order them correctly. Furthermore, the effectiveness of ICA heavily depends on the independence of the columns and the number of columns having non-normal distributions. In most real datasets, these two conditions, independent columns and only one column in normal distribution, are normally unsatisfied, which significantly reduces the effectiveness of ICA, as we show in experiments.

Correspondingly, in the RASP perturbation, we use the following methods to further enhance the resilience to the ICA-based attack: (1) generating the noise component with the normal distribution, (2) using the OPE scheme to convert the column distributions to normal distributions. According to the property of OPE that can change the dimensional value distribution arbitrarily while keeping the value orders unchanged [1], we can choose $E_{ope}$ to output values in normal distributions. As ICA fails to distinguish dimensions with normal distributions, this will effectively protect the original data. This enhancement can be optional. The noise $v$ is drawn from a normal distribution that is designed as follows. The idea is the normal distribution should have the majority population positive. Specifically, we can choose a normal distribution with the mean $\mu$, $\mu > 0$, and the standard deviation $\mu/6$. It is easy to prove that over 99% values drawn from the distribution are positive. With these enhancements, the RASP-perturbed data are resilient to ICA-based attacks.

**Known Input/Output Attack.** With the level 3 knowledge, the attacker knows a number of input/output record pairs. Concretely, let $P_{d \times m}$ be the known $m d$-dimensional original records $(y_1, \ldots, y_m)$, and $Q_{d+2 \times m}$ be the corresponding perturbed $m d + 2$-dimensional
records \((x_1, \ldots, x_m)\). The known input/output attack can be treated as a general regression problem, where the training dataset consists of the tuples \((x_i, y_{ij})\), \(i = 1 \ldots m\) for the \(j\)-th dimension. The attacker tries to learn a model between the feature vectors \(\{x_i, x_i \in \mathbb{R}^{d+2}\}\) and the responses \(\{y_{ij}, y_{ij} \in \mathbb{R}\}\) [8]. In this way, we can build a model for each dimension \(j\).

To understand the effectiveness of the regression-based attack, we analyze a simplified case, where the transformation \(E_{ope}\) is not applied: \(F(x) = A(x^T, 1, v)^T\). This analysis will give a lower-bound accuracy that the known input/output attack can achieve, because the inclusion of \(E_{ope}\) will not make the attack easier.

Let \(A\) decomposed into blocks \(A = (A_1, A_2, A_3)\), where \(A_1\), \(A_2\) and \(A_3\) have block sizes \((d+2) \times d\), \((d+2) \times 1\) and \((d+2) \times 1\), respectively, and the extended data be

\[
\begin{pmatrix}
P \\ I \\ V
\end{pmatrix}
\]

where \(I\) is the row vector with ‘1’ and \(V\) is a row vector with random positive values, corresponding to the two additional dimensions in RASP. Thus, the relationship between \(P\) and \(Q\) can be represented as

\[
Q = (A_1, A_2, A_3) \begin{pmatrix} P \\ I \\ V \end{pmatrix} = A_1 P + A_2 I + A_3 V, \tag{4}
\]

where \(A_2 I\) is a translation matrix that adds the vector \(A_2\) to each of the column vectors in \(A_1 P\), and \(A_3 V\) is a random noise matrix. With sufficient number of known record pairs \((m > d + 2)\), the translation component can be canceled out; then, the regression method can be applied to estimate \(A_1\). With the estimate \(\hat{A}_1\) of \(A_1\), \(A_2\) can be estimated as well. Therefore, for a perturbed dataset \(Y\), the estimate of the original data \(X\) is

\[
\hat{X} = (\hat{A}_1^T \hat{A}_1)^{-1} \hat{A}_1^T (Y - \hat{A}_2 I),
\]

according to the regression theory.

Now we analyze the error of this estimation, which gives a lower bound of the MSE for the general cases. Let \(H = (A_1^T A_1)^{-1} A_1^T\) and \(\hat{H}\) be the version with \(\hat{A}_1\). Let \(H_j\) represent the
According to the definition of RASP, each dimension $X_j$ is generated by $X_j = H_j(Y - A_2I - A_3V)$. With the estimate $\hat{A}_1$ and $\hat{A}_2$, $\hat{X}_j = \hat{H}_j(Y - \hat{A}_2I)$. Then, we have

$$\hat{X}_j - X_j = (\hat{H}_j - H_j)Y + (H_jA_2I - \hat{H}_j\hat{A}_2I) + H_jA_3V.$$  \hspace{1cm} (5)

It follows that

$$MSE(X_j, \hat{X}_j) = var(\hat{X}_j - X_j) > var(H_jA_3V) = (H_jA_3)^2\sigma_{RG}^2,$$  \hspace{1cm} (6)

where $H_jA_3$ is a constant determined by the matrix $A$, and $\sigma_{RG}^2$ is the variance of the noise. Therefore, as long as we make sure the matrix $A$ and the noise generator are appropriately selected so that the values $H_jA_3$ and $\sigma_{RG}^2$ are sufficiently large, the perturbed dataset is resilient to the known input/output attack. One of the RASP’s unique advantages is that the increase of noise variance does not affect the utility of the perturbed data. We will show this property in query processing.

### 3 Range Query Processing with RASP

Based on the RASP encryption we proposed, we target to provide two kinds of query services: range query and k nearest neighbors. Since RASP encryption is convexity preserving and a range query can represented as a convex set query, in the encrypted space there is a unique convex set that is the answer to the query. To efficiently process range query, we could use multidimensional index trees, such as R-Tree [16] that handles axis-aligned minimum bounding boxes (MBR). However, if we still depend on the multidimensional indexing to process the transformed queries, the processing algorithm should be slightly modified to handle arbitrary convex areas because the boundaries of transformed queries are in arbitrary convex shape, not necessarily axis-aligned as Figure 2 shows. So we use a two-stage query processing strategy to process the encrypted query. In the first stage, the proxy trans-
forms the original range space to a polyhedron and finds the MBR of polyhedron using vertex-based algorithm. To submit a range query for the MBR of this polyhedron, we could get the initial result set. In the second stage, we could linear scan the data in the initial result set and filter out the data which are not in the polyhedron. In the following sections, we will introduce a simple query transformation first and then convert it into one against query-based attacks.

### 3.1 Query Transformation

In this section, we discuss how to transform an original range query into the encrypted one in details. First, let’s look at the general form of a range query condition. Let \( X_i \) be an attribute in the database. A simple condition in a range query involves only one attribute and is of the form “\( X_i \ op \ a_i \)”, where \( a_i \) is a constant in the normalized domain of \( X_i \) and \( op \in \{<, >, =, \leq, \geq, \neq\} \) is a comparison operator. For convenience we will only discuss how to process \( X_i \leq a_i \), while the proposed method can be slightly changed for other conditions. Any complicated range query can be transformed into the disjunction of a set of conjunctions, i.e., \( \bigcup_{j=1}^{n} (\bigcap_{i=1}^{m} C_{i,j}) \), where \( m, n \) are some integers depending on the original query conditions and \( C_{i,j} \) is a simple condition about \( X_i \). Again, to simplify the presentation we restrict our discussion to single conjunction condition \( \cap_{i=1}^{m} C_i \). A simple condition \( X_i \leq a_i \) is a halfspace condition. Following the previous discussion, \( X_i \leq a_i \) is converted to the extended vector representation first: \( u^T z \leq 0 \), where \( u \) is a \( d+2 \) dimensional vector with \( u_i = 1, u_{k+1} = -a_i, \) and \( u_j = 0 \) for \( j \neq i, d+1 \), (for \( X_i \geq a_i \), \( u_i = -1, u_{d+1} = a_i \)), and \( z = (x^T, 1, v) \). Then, let \( y \) be the auxiliary vector, i.e., \( y = Az \). The original condition is transformed to the form of Eq. 3 in the encrypted space.

As Proposition 4 shows, searching with the transformed queries on the auxiliary vectors is equivalent to searching with the original queries and data. Someone may notice that this simple query transformation is vulnerable to query-based attacks if the attacker is able to know original query and the transformed query at the same time. In next section we discuss
3.2 Secure Query Transformation Countering Query-based Attacks

To counter against query-based attacks, we have to construct secure query conditions. The additional dimension $x_{d+2}$ is used to construct secure query condition with the simple query transformation. Using the extended vector form $z^T = (x^T, 1, v)$, we have $X_i - a_i = z^T u$ and $X_{d+2} = w^T z$, where $u_i = 1$, $u_{d+1} = -a_i$, $u_j = 0$, for $j \neq i, d + 1$; $w_{d+2} = 1$ and $w_j = 0$, for $j \neq d + 2$. With the transformation $y = Az$, we get the transformed quadratic query condition

$$y^T(A^{-1})^Tuw^TA^{-1}y \leq 0.$$  \hspace{1cm} (7)

Let $\Theta = (A^{-1})^Tuw^TA^{-1}$. In the two-stage processing strategy, the bounding box of the transformed query area is calculated in the proxy server as we discussed earlier. Then, this bounding box and the parameters $\Theta_i$ for each condition $i$, are submitted to the server. Thus, assume each dimension is represented with two half space conditions, the encrypted query $E_Q$ is represented as $\{ \text{MBR}, \{\Theta_1, \ldots, \Theta_{2d}\} \}$. The server will use the bounding box to get the first-stage results and then use the conditions, e.g., $y^T\Theta_i y \leq 0$, to filter out the results.
4 KNN-R: Using Range Queries to Process kNN Queries

As we have mentioned, the quality of secure outsourced kNN query service can be summarized as the CPEL criteria: data Confidentiality, query Privacy, Efficient query processing, and Low inhouse workload. The proposed kNN-R approach aims to achieve a balanced overall CPEL criteria. In the following discussion, we use the “client” to represent the data owner’s inhouse proxy server.

The confidentiality of data and the query privacy are guaranteed by using the RASP encryption. We will discuss the security issues in next section. The server-side efficiency is achieved by the index-aided query processing and an efficient secure binary range query algorithm. Low client-side cost is achieved by fast query preprocessing and high precision query results. In this section, first, we will give the overview of the kNN-R algorithm and then discuss the server-side and client-side components of the query processing algorithm.

4.1 Overview of the kNN-R Algorithm

The kNN-R algorithm aims to improve the confidentiality guarantee while preserving the efficiency of query processing. The basic idea is to use the RASP encryption to protect the confidentiality of data, and to use secure range query to protect the privacy of kNN query. The key is to develop an efficient kNN query algorithm based on the RASP encrypted data and queries.

The design of kNN-R algorithm keeps the following problems in mind. (1) While the RASP protects the data confidentiality, it does not preserve distances or distance ranks. Therefore, the traditional distance-based kNN search algorithm does not work with the RASP encrypted data. Can we design a kNN search algorithm based on existing RASP range query algorithm? (2) Because of the limited computing capacity of the client side (the proxy server in Figure 1), the new algorithm should minimize the client’s responsibility in query processing, which includes pre-processing, post-processing, and in-processing aid. Thus, the second question is how we design the algorithms to minimize the client’s costs.
Figure 3: Procedure of KNN-R algorithm

Figure 4: Illustration for kNN-R Algorithm when k=3
We briefly describe the kNN-R algorithm as follows. In the kNN-R algorithm the client and the server collaboratively finish the kNN query processing. This algorithm is based on square ranges to approximately find the kNN candidates for a query point. We define square range as follows.

**Definition 3.** A square range is a hyper cube that is centered at the query point and has equal-length edges.

The basic idea is to find the square range that contains at least k points and then expands it to surely include the kNN candidates. Figure 4 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** is the square range extended from the inner range. The outer range surely contains the kNN results (Proposition 5) but it may also contain irrelevant points that need to be filtered out.

Concretely, the kNN-R algorithm consists of five steps involving both the client and the server. Figure 3 demonstrates the whole procedure of the algorithm and the Algorithm 1 explains the algorithm step by step. The client will generate the initial upper bound range (that contains more than k points) and the lower bound range (that contains less than k points) and send them to the server. The server finds the inner range and returns to the client. 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. The client decrypts the records and pick the top k candidates as the final result.

**Algorithm 1** KNN-R algorithm

1: The client generates the initial range and sends its secure form to the server;
2: The server works on the secure range queries and finds the inner range covering at least k points;
3: The client decodes the secure inner range from the server and extends it to the outer range, which is sent back to the server;
4: The server returns the points in the outer range
5: The client decrypts the points and extracts the k nearest points;

It is very easy to verify that the following conclusion.
Proposition 5. The kNN-R algorithm returns results with 100% recall.

Sketch of Proof: The sphere in Figure 4 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.

If the points are approximately uniformly distributed, we can estimate the precision of the server returned results. With the assumption of uniformly distributed points, the number of points contained by the inner range and the outer range is proportion to their areas. Based on the relationship between them, it is easy to derive that the area of outer range is two times of the inner range. Thus, if the inner range contains $m$ points, $m \geq k$, the number of points contained by the outer range is around $2m$ points. The precision is $\frac{k}{2m}$. Therefore, we know the precision for uniformly distributed data is around 0.5.

There are three challenges in this process. (1) Because the inner range will determine the precision of the result, we want to find inner ranges as compact as possible. (2) As the server works on encrypted range queries only, it is challenging to find the inner range based on encrypted range queries. Is it possible that the server works on the initial ranges solely to derive a compact inner range without additional interactions with the client? (3) The setting of initial range may also affect the efficiency of server-side processing. Can the client quickly decide a valid initial range?

4.2 Definitions of Compact Inner Square Range

An important step in the kNN-R algorithm is to find the compact inner square range to achieve high precision. However, finding compact inner range may involve a complicated tradeoff between the precision and the cost of query processing, which in turn implies a tradeoff between the client’s load and the server’s load. To study this tradeoff, in the following, we give two definitions of compact inner square range: the $k$ minimum range and the $(k, \delta)$-range. Both definitions will guarantee 100% recall.
Let $S_{-\epsilon}$ represent that each side of a square range $S$ shrinks by $\epsilon$ distance to the center. We have the following definition.

**Definition 4.** A square range, $S$, centered at the query point is the $k$ minimum range if and only if $S$ covers at least $k$ points, but for any small number $\epsilon$, $S_{-\epsilon}$ covers less than $k$ points.

The $k$ minimum range defines the smallest inner square range that satisfies the requirement of Algorithm 1 to ensure 100\% recall. Because of the precise definition, there is a substantial cost to achieve the $k$ minimum range. On the other hand, if we consider the balance between the cost and the precision, we may need a more flexible definition. We give a relaxed definition $(k, \delta)$-range as follows.

**Definition 5.** A $(k, \delta)$-range is any square range centered at the query point, the number of points in which is in the range $[k, k + \delta]$, $\delta$ is a nonnegative integer.

We will study the algorithms to efficiently find these ranges and investigate the relationship and tradeoffs between the precision and the efficiency of query processing.

### 4.3 Algorithms for Finding Inner Square Range

With the precise definitions of the $k$ minimum range and the $(k, \delta)$-range, we design algorithms to find them, respectively. The basic algorithms are based on binary range search.

**K Minimum Range Algorithm**

Suppose a search space 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 $S^{(in)} \subset S^{(out)}$. It directly follows that $N^{(in)} \leq N^{(out)}$. It is also easy to derive that

**Corollary 1.** If $N^{(in)} < N^{(out)}$, $S^{(in)} \subset S^{(out)}$. 

Using this definition, we can always construct a series of enclosed square ranges centered on the query point: \( S^{(L_1)} \subset S^{(L_2)} \subset \ldots \subset S^{(L_m)} \). Correspondingly, the numbers of points enclosed by \( \{ S^{(L_i)} \} \) have the ordering \( N^{(L_1)} \leq N^{(L_2)} \leq \ldots \leq N^{(L_m)} \). Assume \( S^{(L_1)} \) is the range containing only the query point, and \( S^{(L_m)} \) is the initial upper bound range returned by the client. The problem of finding the \( k \) minimum range \( S \) can be mapped to the binary search over the sequence \( \{ S^{(L_i)} \} \).

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 \( N^{(low)} < k \leq N^{(high)} \) in each step, which is achieved with the following procedure. First, we find the middle square range \( S^{(mid)} \), where and \( mid = (low + high) / 2 \). 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 \( S^{(L_1)} \) and \( S^{(high)} \) is \( S^{(L_m)} \). Afterwards this binary search process repeats until \( high - low < \mathcal{E} \), where \( \mathcal{E} \) is a predefined small value. Algorithm 2 describes these steps.

Algorithm 2 K Minimum Range Algorithm

1: procedure K-MINIMUM-RANGE\((L_1, L_m, k)\)
2:  \( high \leftarrow L_m, low \leftarrow L_1; \)
3:  while \( high - low \geq \mathcal{E} \) do
4:    \( mid \leftarrow (high + low) / 2; \)
5:    num \leftarrow number of points in \( S^{(mid)} \);
6:    if num \( \geq k \) then
7:      \( high \leftarrow mid; \)
8:    else
9:      \( low \leftarrow mid; \)
10:  end if
11: end while
12: return \( S^{(high)} \);
13: end procedure

Let’s define the distance between \( S^{(L_i)} \) and \( S^{(L_j)} \) as

\[
D(S^{(L_i)}, S^{(L_j)}) = |L_i - L_j|.
\]

Each step of the loop results in a reduced distance \( \alpha_i \), of which \( \alpha_0 = |L_m - L_1| \), and \( \alpha_i = \alpha_{i-1} / 2 \). When \( \alpha_i < \mathcal{E} \), the algorithm stops. We have the following proposition.
**Proposition 6.** The output of Algorithm 2 is a square range within $\mathcal{E}$ distance to the $k$ minimum range.

**Sketch of Proof:** Note that the statements 6-10 in Algorithm 2 guarantee $N^{(low)} < k \leq N^{(high)}$ in each update. With Corollary 1 and the definition of $k$ minimum range, we have $S^{(low)} \subset S \subset S^{(high)}$. It follows that $D(S^{(high)}, S) < D(S^{(high)}, S^{(low)})$. Once the distance $\alpha_i < \mathcal{E}$, we have $D(S^{(high)}, S) < D(S^{(high)}, S^{(low)}) = \alpha_i < \mathcal{E}$. □

**Complexity.** The algorithm results in a series of distances, $\alpha_i$, between the higher and the lower bounds. The initial distance $\alpha_0 = |L_m - L_1|$ and sequentially $\alpha_i = \alpha_{i-1}/2$. When $\alpha_i < \mathcal{E}$, the algorithm stops. Let the total number of iterations be $r$. It follows $\alpha_r = \alpha_0/2^r < \mathcal{E}$. Therefore, the number of iteration is

$$r = \lceil \log \frac{\alpha_0}{\mathcal{E}} \rceil. \quad (8)$$

It is related to the selections of $S^{(L_m)}$ and $S^{L_1}$, and also the setting of $\mathcal{E}$.

**(k, $\delta$)-Range Algorithm**

According to Equation 8, by tuning $\mathcal{E}$ we can possibly make a tradeoff between the precision and the performance. Alternatively, we can find the $(k, \delta)$-range where $\delta$ is tunable. We slightly change Algorithm 2 to derive an algorithm for finding $(k, \delta)$-range. Because $(k, \delta)$-range is any square range of which the number of points is between $k$ and $k+\delta$, as long as we encounter such a square range in the binary search, the algorithm can stop. Algorithm 3 gives the details.

There is an intricate relationship between the $k$ minimum range algorithm and the $(k, \delta)$-range algorithm. Assume the point distribution is uniform. Let the length of the $k$ minimum range be $a$, and that of the square range containing $k+\delta$ points be $b$. It is easy to obtain that $b \approx \sqrt{\frac{k+\delta}{k}} a$. Therefore, when we set $\mathcal{E} = (b - a)/2 \approx 1/2(\sqrt{\frac{k+\delta}{k}} - 1)a$, the $k$ minimum range algorithm returns approximately the same result as the $(k, \delta)$-range algorithm does.
Algorithm 3 \((K, \delta)-\text{Range Algorithm}\)

1: \textbf{procedure} \((K, \delta)-\text{RANGE}(L_1, L_m, k, \delta)\)
2: \hspace{1em} high \leftarrow L_m, low \leftarrow L_1;
3: \hspace{1em} \textbf{while} high − low \geq \mathcal{E} \hspace{1em} \textbf{do}
4: \hspace{2em} mid \leftarrow (\text{high} + \text{low})/2;
5: \hspace{2em} num \leftarrow \text{number of points in } S^{(\text{mid})};
6: \hspace{2em} \textbf{if} num \geq k \&\& num \leq k + \delta \hspace{1em} \textbf{then}
7: \hspace{3em} \text{Break the loop;}
8: \hspace{2em} \textbf{else if} num > k + \text{delta} \hspace{1em} \textbf{then}
9: \hspace{3em} \text{high} \leftarrow \text{mid};
10: \hspace{2em} \textbf{else}
11: \hspace{3em} \text{low} \leftarrow \text{mid};
12: \hspace{2em} \textbf{end if}
13: \hspace{1em} \textbf{end while}
14: \hspace{1em} \textbf{return} S^{(\text{mid})};
15: \textbf{end procedure}

However, since the distribution is often non-uniform, the \((k, \delta)\)-range algorithm is more flexible and efficient than \(k\) minimum algorithm. We will compare them in experiments.

### 4.4 Finding Inner Range with Secure Range Queries

Algorithm 2 and 3 give the basic ideas of finding the inner range. There are two critical operations in these algorithms: (1) finding the number of points in a square range and (2) updating the higher and lower bounds. Under the secure range query framework, the first operation can be securely conducted by the server. However, it is challenging to efficiently implement the second operation. In this section, we first describe a straightforward solution, which, however, needs multiple rounds of interactions between the client and the server, thus too expensive to be practical. Then, we give an algorithm that allows the server to work on the secure higher and lower bound range queries to derive the secure middle range query without the client’s help, which significantly improves the efficiency of server-side query processing.
Client-Server Interaction Method

Because the server does not know the plain value of high and low, the calculation: \( \text{mid} = (\text{high} + \text{low})/2 \) and the update to the higher or lower bound have to be done in the client. Concretely, if we look at the k minimum range algorithm, Algorithm 2, the client has to execute all steps except step 5, which is done by submitting a range query for \( S^{(\text{mid})} \) to the server to find the number of enclosed points. As we have mentioned in Section 4.3, the number of interactions will be \( O(\log \frac{\alpha_0}{\epsilon}) \) for the k minimum range algorithm, which could be expensive. The \((k, \delta)\)-range algorithm needs less number of interactions, but it would be better to eliminate all the interactions. The following algorithm describes a server-only solution.

Server Only Method

The client-server interaction method requires a significant amount of communication cost. It is better to eliminate or reduce this cost. In the following, we show that it is possible to compute the encrypted middle range based on encrypted higher and lower ranges, if they are encrypted with the RASP encryption.

As discussed in the RASP query processing [4], a range query \( S^{(L)} \) is securely encoded as the MBR\((L)\) of its polyhedron range in the encrypted space and the \(2(d + 2)\) dimensional conditions\(^1\) \( y^T \Theta_i^{(L)} y \leq 0 \) determining the sides of the polyhedron, each of the \(d + 2\) extended dimensions gets a pair of conditions for the upper and lower bounds, respectively. The problem of binary search is to use the higher bound range \( S^{(\text{high})} \) and the lower bound range \( S^{(\text{low})} \) to derive \( S^{(\text{mid})} \). When all of these ranges are encrypted, the problem is transformed to (1) deriving \( \Theta_i^{(\text{mid})} \) from \( \Theta_i^{(\text{high})} \) and \( \Theta_i^{(\text{low})} \); and (1) deriving MBR\((\text{mid})\) from MBR\((\text{high})\) and MBR\((\text{low})\).

Finding \( \Theta_i^{(\text{mid})} \). Without loss of generality, we consider only the upper bound condition for dimension \( i \) - the lower bound can be handled similarly. Let \( X_i \leq s_{i, \text{max}} \) be the upper bound

\(^1\)For the extended \(d+2\) dimensions, including the ‘1’ dimension and the noise dimension.
condition. Correspondingly, \(s_{i,\text{max}}^{(\text{high})}, s_{i,\text{max}}^{(\text{low})}\), and \(s_{i,\text{max}}^{(\text{mid})}\) are the upper bounds for the ranges \(S^{(\text{high})}, S^{(\text{low})}\), and \(S^{(\text{mid})}\). According to the algorithm, there is a relationship \(s_{i,\text{max}}^{(\text{mid})} = (s_{i,\text{max}}^{(\text{high})} + s_{i,\text{max}}^{(\text{low})})/2\).

In RASP, the condition \(X_i \leq s_{i,\text{max}}\) is equivalent to \((X_i - s_{i,\text{max}})X_{k+2} \leq 0\) because of \(X_{k+2} > 0\). \((X_i - a_i)X_{k+2} \leq 0\) is further transformed to the encrypted form

\[
y^T(A^{-1})^Tuw^TzA^{-1}y \leq 0. \tag{9}
\]

where for the vector \(u\), \(u_i = 1\), \(u_{k+1} = -s_{i,\text{max}}\), \(u_j = 0\), for \(j \neq i, k + 1\); for the vector \(w\), \(w_{k+2} = 1\) and \(w_j = 0\), for \(j \neq k + 2\). We use \(\Theta_i\) to denote the parameter matrix \((A^{-1})^Tuw^TzA^{-1}\).

Let’s represent \(A^{-1}\) with block matrices: \(A^{-1} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}\), \(A_1\) is the first \(d\) rows of \(A^{-1}\), a \(d \times (d + 2)\) matrix, and \(A_2\) is the last two rows of \(A^{-1}\), a \(2 \times (d + 2)\) matrix. \(w^T\) is represented as \(w^T = \begin{bmatrix} 0 \\ I_2 \end{bmatrix}\), where \(0\) is a zero vector of length \(d\) and \(I_2 = \begin{bmatrix} 0 & -1 \end{bmatrix}\).

Let the vector \(u\) be \(u = \begin{bmatrix} I_1 \\ B \end{bmatrix}\), where \(I_1\) is a \(d \times 1\) vector with all elements zero except its \(i\)-th element is 1 and \(B = \begin{bmatrix} -s_{i,\text{max}} \\ 0 \end{bmatrix}\). With all these block matrix representation, \(\Theta_i = \begin{bmatrix} A_1^T \\ A_2^T \end{bmatrix}\begin{bmatrix} I_1 \\ B \end{bmatrix}\begin{bmatrix} 0 & I_2 \end{bmatrix}\)

\[
= \begin{bmatrix} A_1^TI_1 + A_2^TB \\ I_2A_2 \end{bmatrix} = A_1^TI_1I_2A_2 + A_2^TBI_2A_2. \tag{10}
\]

Note that \(A_1^TI_1I_2A_2\) is a constant matrix, independent of the condition. We denote the constant part with \(P_1 = A_1^TI_1I_2A_2\).

Let \(P_2\) be the remaining part \(P_2 = A_2^TBI_2A_2\).

Let \(A_2\) be \(A_2 = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,d+2} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,d+2} \end{bmatrix}\).
We have

\[
P_2 = A_2^T \begin{bmatrix} -s_{i,\text{max}} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \end{bmatrix} A_2
\]

\[
= \begin{bmatrix} a_{1,1} & a_{2,1} \\ a_{1,2} & a_{2,2} \\ \vdots & \vdots \\ a_{1,d+2} & a_{2,d+2} \end{bmatrix} \begin{bmatrix} 0 & -s_{i,\text{max}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_{1,1} & \cdots & a_{1,d+2} \\ a_{2,1} & \cdots & a_{2,d+2} \end{bmatrix}
\]

\[
= -s_{i,\text{max}} \begin{bmatrix} a_{1,1} & a_{2,1} \\ a_{1,2} & a_{2,2} \\ \vdots & \vdots \\ a_{1,d+2} & a_{2,d+2} \end{bmatrix} \begin{bmatrix} a_{2,1} & a_{2,2} & \cdots & a_{2,d+2} \\ 0 & 0 & \cdots & 0 \end{bmatrix}
\]

Thus, we can represent \( P_2 \) with \(-s_{i,\text{max}}P_3\), where \( P_3 \) is the constant part in the above equation. It follows that \( \Theta_i = P_1 - s_{i,\text{max}}P_3 \), and we can derive

\[
(\Theta_i^{(\text{high})} + \Theta_i^{(\text{low})})/2 = P_1 - (s_{i,\text{max}}^{(\text{high})} + s_{i,\text{max}}^{(\text{low})})/2 \times P_3 = \Theta_i^{(\text{mid})}.
\]

It shows that \( \Theta_i^{(\text{mid})} \) can be directly computed with \( \Theta_i^{(\text{high})} \) and \( \Theta_i^{(\text{low})} \).

**Finding \( MBR^{(\text{mid})} \).** In general, the MBR of an arbitrary polyhedron can be derived based on the vertices of the polyhedron. Based on the property of convexity preserving of RASP [4], a polyhedron is mapped to another polyhedron in the encrypted space. Concretely, let a polyhedron \( P \) has \( m \) vertices \( \{x_1, \ldots, x_m\} \), which are mapped to the vertices in the encrypted space: \( \{y_1, \ldots, y_m\} \). Then, the upper bound and lower bound of dimension \( j \) of the MBR of the polyhedron in the encrypted space are determined by \( \max\{y_{ij}, i = 1 \ldots m\} \) and \( \min\{y_{ij}, i = 1 \ldots m\} \), respectively.

Since we only use MBR to reduce the set of results for filtering, a slightly larger MBR would still guarantee the correctness of the MBR based query processing algorithm, with
possibly increased filtering cost. In the following, we try to find such a MBR to enclose 
MBR\(^{(mid)}\). By the definition of the square ranges \(S^{(low)}\), \(S^{(mid)}\) and \(S^{(high)}\), their vertices have the relationship \(x_i^{(mid)} = (x_i^{(low)} + x_i^{(high)})/2\). The images of the vertices are notated as \(y_i^{(low)}\), \(y_i^{(high)}\), and \(y_i^{(mid)}\), respectively. Correspondingly, the MBR\(^{(mid)}\) in the encrypted space should be found from \(\{y_1^{(mid)}, \ldots, y_m^{(mid)}\}\), where \(y_i^{(mid)} = A(x_i^{(mid)}, 1, v_i^{(mid)})^T\). Since \((y_i^{(low)} + y_i^{(high)})/2 = A(x_i^{(mid)}, 1, (v_i^{(low)} + v_i^{(high)})/2)^T\), and \((v_i^{(low)} + v_i^{(high)})/2\) is a valid positive random number. Thus, MBR\(^{(mid)}\) can be determined with vertices \(\{(y_i^{(low)} + y_i^{(high)})/2\}\).

Let the j-th dimension of MBR\(^{(L)}\) represented as
\[
[s_j^{(L)}^{\text{min}}, s_j^{(L)}^{\text{max}}], \text{ where } s_j^{(L)}^{\text{min}} = \min\{y_{ij}^{(L)}, i = 1 \ldots m\}, \text{ and } s_j^{(L)}^{\text{max}} = \max\{y_{ij}^{(high)}, i = 1 \ldots m\}. \]
Now we choose the MBR\(^{(MID)}\) as follows: for j-th dimension we use \([s_j^{(low)} + s_j^{(high)}]/2, (s_j^{(low)} + s_j^{(high)})/2\]. We show that

**Proposition 7.** MBR\(^{(MID)}\) encloses MBR\(^{(mid)}\).

**Proof.** For two sets of \(m\) real values \(\{a_1, \ldots, a_m\}\) and \(\{b_1, \ldots, b_m\}\), it is easy to verify that

\[
\max\{a_1, \ldots, a_m\} + \max\{b_1, \ldots, b_m\} \geq \max\{a_1 + b_1, \ldots, a_1 + b_m\}
\]

\[
\min\{a_1, \ldots, a_m\} + \min\{b_1, \ldots, b_m\} \leq \min\{a_1 + b_1, \ldots, a_1 + b_m\}.
\]

Thus, \((s_{i,\text{min}}^{(low)} + s_{i,\text{min}}^{(high)})/2 \leq \min\{y_{ij}^{(low)} + y_{ij}^{(high)}\}/2, i = 1 \ldots m\} = s_{i,\text{min}}^{(mid)}, \text{ and } (s_{i,\text{max}}^{(low)} + s_{i,\text{max}}^{(high)})/2 \geq s_{i,\text{max}}^{(mid)}\). Since for each dimension, MBR\(^{(MID)}\) encloses MBR\(^{(mid)}\), we have
MBR\(^{(MID)}\) encloses MBR\(^{(mid)}\). \(\square\)

Since MBR\(^{(MID)}\) can be calculated based on MBR\(^{(high)}\) and MBR\(^{(low)}\), and \(\Theta_i^{(mid)} = (\Theta_i^{(high)} + \Theta_i^{(low)})/2\), the server can process the \(k\) minimum range or (\(k, \delta\))-range algorithm without the help of the client.
4.5 Finding Compact Initial Range with Density Map

The complexity of both the $k$ minimum range algorithm and the $(k, \delta)$-range algorithm is determined by the initial higher bound range provided by the client. Thus, it is important to provide a compact initial range to help the server processes queries more efficiently. A naive method for determining the initial range is to provide the entire domain space. Assume each dimension is bounded, i.e., the dimension $i$ has a bounded domain $[s_{i,\text{min}}, s_{i,\text{max}}]$. Let the query point $Q$ be $Q = (q_1, q_2, \ldots, q_d)$. The problem is to find a sphere centered on the query point $Q$ that covers the entire domain. For the $i$-th dimension, if we draw a circle centered on $q_i$ to cover the whole dimensional domain, the radius $r_i$ is $\max\{q_i - s_{i,\text{min}}, s_{i,\text{max}} - q_i\}$.

For all $d$ dimension, if we draw a sphere to cover the entire $d$ dimensional domain, then the radius $r = \max\{r_1, r_2, \ldots, r_d\}$. Therefore, the initial higher bound range is $[q_i - r, q_i + r]$ for dimension $i$.

We use the density-map method to help the proxy find the compact initial range. A density map is a multidimensional equi-width histogram. We partition each dimension equivalently into $C$ shares to generate a $C^d$-cell grid structure. A $d$-dimensional point is uniquely mapped to one of the cells. Each cell records the number of points mapped to it. We define

**Definition 6.** The distance between a cell and the query point is the distance from the query point to the furthest corner of the cell.

Algorithm 4 shows an example in two dimensional space. It aims to find the smallest circle, in which the completely covered cells contains at least $k$ points. Each time it extends the circle by looking at the closest cell, of which the distance to the query point is defined as Definition 6. The algorithm uses a heap to maintain the closest cell. When a cell is chosen as the closest cell, the distances of its adjacent neighbors which haven’t been inserted in the heap are pushed into the heap. The algorithm keeps extending the circle until the total number of points in the circle is at least $k$ points. For example, in Figure 5 there are four cells completely covered by the circle, which contains $2+1+2+4=9$ points in total. Therefore, this is the smallest circle which definitely covers 9 points based on the density map.
Algorithm 4 Preprocessing of Binary Search

1: procedure FINDING COMAPCT INTIAL RANGE(density map P, Value k)
2: \( H_k \leftarrow \text{new min-heap of pairs } < \text{distance, id} > \)  \( \triangleright \) id is the position of the cell
3: Insert one pair of \( < \text{distance, id}_0 > \)  \( \triangleright \) \( id_0 \) is the id of the cell that covers the query point.
4: \( \text{sum} \leftarrow 0 \)
5: \( \text{radius} \leftarrow 0 \)
6: while \( \text{sum} < k \) do
7: Pop up the top element of the heap \( H_k \)  \( \triangleright \) Top element is the closest cell
8: \( \text{radius} \leftarrow \text{distance of the cell to the query point} \)
9: \( \text{num} \leftarrow \text{number of points of the closest cell} \)
10: \( \text{sum} \leftarrow \text{sum} + \text{num} \)
11: Insert into \( H_k \) for the cells adjacent to the closest cell which have not been inserted into the H before
12: end while
13: return \( \text{radius} \)
14: end procedure

Complexity. In the best case, the algorithm only needs to include one closest cell, which contains more than \( k \) points. In worst case, this algorithm will possibly iterate all cells, when the density map is extremely sparse. If the points are roughly uniformly distributed, with each cell contains \( m \) points, then the average cost will be \( k/m \) iterations.

Figure 5: Calculating Points in the Circle

The initial higher bound square range is defined as the bounding box of the final circle.
4.6 Attacks on $k$NN Queries

In the $k$NN-R algorithm, the proxy server prepares the initial ranges $S^{(L_1)}$ and $S^{(L_m)}$ and submits them to the cloud. A secured range query consists of two parts: the MBR of the transformed polyhedron and the $2(d+2)$ dimensional range conditions $u^T\Theta_i u \leq 0$, $i = 1 \ldots 2(d+2)$. However, one may concern that the MBR of the secured $S^{(L_1)}$, which is a narrow range encoded for the query point, may reveal the query point and the information about the RASP perturbation parameters.

We describe the $S^{(L_1)}$-based attack with the case of one-dimensional data. Assume the query point is $x$, $x \in \mathbb{R}$. It is extended to $(x, 1, v)^T$ with random value $v$ in the range $(0, D)$, and then transformed with an invertible $3 \times 3$ matrix $A$. For simplicity of analysis, we consider that the low-bound range uses the exact value of $x$, which is $S^{(L_1)} = ([x, x], [1, 1], [0, D])$ - in practice, we use $([x - \delta, x + \delta], [1, 1], [0, D])$, where $\delta$ is a small positive value. Such a range is just a line segment in the extended space (Left subfigure of Figure 6). With the transformation $A$, the transformed line segment has a cubic MBR (right subfigure of Figure 6).

Furthermore, we assume that the attacker knows $(x, 1, 0)^T$ is mapped to $\beta$, one of the vertices of the MBR, $\beta = (b_1, b_2, b_3)^T$. And the attacker manages to get a number of such pairs: $(x_i, 1, 0)^T \rightarrow \beta_i$ for $i = 1 \ldots m$ (corresponding to the level 3 knowledge). Let the first row of $A$ be $A_1$, $X = ((x_1, 1, 0)^T, \ldots, (x_m, 1, 0)^T)$, and $B = (\beta_1^T, \ldots, \beta_m^T)$. Let $B_1$ be the first row of $B$. Then, we have $A_1X = B_1$. The key is whether the attacker can solve the linear equation to find $A_1$. 
We show that the attacker is impossible to solve the linear equation. Because $X$ has the rows of constant ‘1’ and ‘0’, its rank $\leq 1$. Based on the rank-nullity theorem [17], the solution to $A_1X = 0$ is $A_1 = a_1u_1^T + a_2u_2^T$, where $u_1$ and $u_2$ are the basis of the null space of $X$; $a_1$ and $a_2$ can be any random numbers. Thus, the attacker cannot determine a unique solution for $A_1X = B_1$. This analysis shows that the RASP perturbation is also resilient to the attack on the MBRs of narrow ranges.

5 Experiments

In this section, we present four sets of experimental results to show the unique features of the proposed kNN-R approach: (1) Comparative study on the $k$ minimum range algorithm and the $(k, \delta)$-range algorithm; (2) Cost reduction brought by the preprocessing algorithm; (3) overall performance of query processing and the relationship between the APC setting and performance; (4) Performance and precision advantages over the Casper approach [19] and the dot-product based kNN approach [22].

5.1 Setup

Two datasets are used in experiments: (1) a synthetic multidimensional dataset that draws 100,000 samples uniformly at random from the domain $[0, 50000]$ for each dimension; (2) North East dataset, that contains 123593 two dimensional addresses, downloaded from the RTree Portal website (http://www.rtreeportal.org/). In order to index encrypted data, we use an R*-tree implementation provided by Dr. Hadjieleftheriou at AT&T Lab. We set the block size to 4KB and 100 records per block. In each individual experiment we generate 1000 queries uniformly at random from the domain. All experiments are performed in a PC with an Intel i5 CPU and 6GB RAM. The dot-product approach [22] has to scan the entire database to find kNN, while our approach can utilize the index built on the encrypted data for range queries. If we use the
number of block accesses of the database to represent the cost of query, the cost of the dot-product approach would be proportional to the total number of data blocks in the database. The Casper approach uses cloaking boxes to hide both the original data points in the database and the query points. Based on the description in the paper [19], we implement the 1NN query processing algorithm for the experiment.

5.2 Scalability of Proxy Server

Table 1 shows that in contrast with the service provider, the time cost from the proxy server is much smaller and more scalable if the number of queries is increased. So it is reasonable to provide a proxy server in the system architecture without influencing the scalability of the system.

<table>
<thead>
<tr>
<th>No. of queries</th>
<th>1K</th>
<th>3K</th>
<th>5K</th>
<th>7K</th>
<th>9K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy for Uniform Data</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.04</td>
<td>0.08</td>
</tr>
<tr>
<td>Proxy for Real Data</td>
<td>0.01</td>
<td>0.05</td>
<td>0.05</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>Cloud Server for Uniform Data</td>
<td>1.01</td>
<td>3.09</td>
<td>5.34</td>
<td>6.58</td>
<td>8.77</td>
</tr>
<tr>
<td>Cloud Server for Real Data</td>
<td>1.33</td>
<td>3.98</td>
<td>6.43</td>
<td>8.76</td>
<td>12.28</td>
</tr>
</tbody>
</table>

Table 1: Comparison between Proxy and Server

5.3 k Minimum Range and (k, δ)-Range Algorithms

The $E$ parameter in the $k$ minimum range algorithm and the $\delta$ parameter in the $(k, \delta)$-range algorithm are the parameters determining the tradeoff between the performance and precision. By increasing $E$ or $\delta$, the algorithms needs less rounds to finish, while the precision of the precision decreases. In this set of experiments, we want to understand how the setting of these parameters affects the performance and the result precision.

Figure 7 shows the effect of $E$ setting to the k minimum range algorithm. When $E$ is smaller than 1 (or 1/50000 = 0.002% of the domain size), the precision doesn’t change much - it simply wastes the time of server processing. The optimal precisions are around 0.5, matching with our analysis in Section 4. When $E$ is larger than 1, the precision starts dropping. With $E = 100$ or 0.2% of the domain size, the precision is reduced by about a half of the
optimal precision. On the other hand the average number of rounds in binary search steadily decreases as $E$ increases. The optimal setting seems at $E = 10$, where the precision does not reduce much but the cost is reduced to less than 10 rounds.

Figure 8 shows the effect of $\delta$ setting to the $(k, \delta)$-range algorithm. As $\delta$ becomes larger, the precision and the number of rounds decrease. At $\delta = 0$ the precision and the performance are very similar to the setting of $E = 10$ for the k minimum range algorithm. Comparing these two algorithms, it might be tricky to select appropriate $E$ to avoid wasting server processing resources, while it is easier to set $\delta$ for the $(k, \delta)$-range algorithm.
5.4 Cost Reduction with Preprocessing

Another key factor for the inner range search algorithms is the initial higher bound range $S_{L_m}$. The naive method as we mentioned in Section 4.5 is to use a square range covering the entire domain space. In comparison, our preprocessing algorithm will reduce the range significantly and thus decrease the cost of query processing.

To clearly present the results, we use the rate between the length of the initial range $S_{L_m}$ and the entire domain to represent the size of $S_{L_m}$. And we use the log number of block accesses ($\log_{10}(\text{blocks})$) and the wall clock time (Time) to represent the cost of the k minimum range algorithm. Table 2 shows the numbers based on the average over 1000 random queries on the database of 100 thousands of points. The initial range is dramatically reduced from 100% to 3%; the number of block accesses and the server processing time are reduced in magnitudes.

<table>
<thead>
<tr>
<th>Data&amp; setting</th>
<th>$S_{L_m}$ size</th>
<th>$\log_{10}(\text{blocks})$</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform/no preprocessing</td>
<td>100%</td>
<td>7.70</td>
<td>79</td>
</tr>
<tr>
<td>Uniform/preprocessing</td>
<td>3%</td>
<td>5.57</td>
<td>1.35</td>
</tr>
<tr>
<td>NorthEast/no preprocessing</td>
<td>100%</td>
<td>8.00</td>
<td>141</td>
</tr>
<tr>
<td>NorthEast/preprocessing</td>
<td>3%</td>
<td>5.83</td>
<td>1.70</td>
</tr>
</tbody>
</table>

Table 2: Cost reduction brought by the preprocessing algorithm.

5.5 kNN-R vs Linear Scan Approaches

Many secure approaches cannot use indices for query processing, which results in very poor performance. For example, dot-product approach [22] encodes the points with random projections. It tries to recover dot-products in query processing for distance comparison. The way it encodes data disallows the index-based query processing. Without the aid of indices, processing a kNN query will have to scan the entire database. We compare the number of block accesses and the server times to show that our approach is much more efficient especially handling very large databases because the underlying RASP approach can use indices in query processing.

We use the database of 100 thousands of data points and 1000 randomly selected queries.
for this experiment. Similarly, we use the log number of block accesses (log10(blocks)) and the wall clock time (Time) to represent the cost in Table ??.

The kNN-R results are from the experiments of k minimum range algorithm. The linear scan works on plaintext data, which just gives a rough comparison to show how large the performance difference can be.

<table>
<thead>
<tr>
<th>Data&amp; setting</th>
<th>log10(blocks)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform/Linear Scan</td>
<td>9.06</td>
<td>195</td>
</tr>
<tr>
<td>Uniform/kNN-R</td>
<td>5.57</td>
<td>1.35</td>
</tr>
<tr>
<td>NorthEast/Linear Scan</td>
<td>9.06</td>
<td>195</td>
</tr>
<tr>
<td>NorthEast/kNN-R</td>
<td>5.83</td>
<td>1.70</td>
</tr>
</tbody>
</table>

Table 3: Performance comparison on linear scan and index-aided kNN-R processing.

### 5.6 Comparing kNN-R with the Casper Approach

In this set of experiments, we compare our approach with the Casper approach with a focus on the tradeoff between the data confidentiality and the query result precision (which indicates the workload of the inhouse proxy). We will formally define the amount of confidentiality preserved by the Casper approach based on the size of cloaking box, and then show the relationship between the preserved confidentiality and the precision of query result.

The Casper approach [19] uses cloaking boxes to hide the exact values of the data points and the query points. To find the nearest neighbor the Casper algorithm aims to find an extended area to guarantee that the points in this area must contain the nearest neighbor. Because the extended area has some irrelevant points as well as our method does, the most important comparison to evaluate these two methods is the precision.

Data confidentiality in the Casper approach is guaranteed by the size of cloaking box, which, however, affects the efficiency of query processing and the precision of query results. For the convenience of presentation, we assume each dimension has the same length of domain and each cloaking box is square box. We can conveniently represent the size of cloaking box with the rate, $r$, between the edge length of the cloaking box and the size of the entire domain. Next, we try to derive the confidentiality with the confidentiality measure APC defined in Section 2.4. Thus, if the length of domain is $D$ and the center of
the cloaking box at dimension $i$ is $C$, the $i$-th dimension value $x$ is uniformly distributed in $[C - \frac{rD}{2}, C + \frac{rD}{2}]$. Furthermore, the attacker can only randomly guess the original point based on the cloaking box. Therefore, the random guess $\hat{x}$ has the same uniform distribution over $[C - \frac{rD}{2}, C + \frac{rD}{2}]$. Thus, the difference $x - \hat{x}$ follows a so-called triangle distribution \([\cdot]\) over $[C - rD, C + rD]$, of which the variance is $\frac{r^2D^2}{6}$, i.e., the standard deviation $\sigma = \frac{rD}{\sqrt{6}}$. For normalized distribution of $x - \hat{x}$ (triangle distribution over $[-\sqrt{6}, \sqrt{6}]$), we can choose $\delta = \sqrt{6}$ to cover the entire domain with confidence 1. Therefore, we have

$$APC_{\text{Casper}, r} = 2\delta\sigma/D = 2r,$$

with confidence 1 in terms of the random guess attack.

Figure 9 shows that when the Casper’s APC, i.e., $2r$ is increased from 0.4% to 2%, the precision dramatically drops from 62% to 13% for the uniform data and 43% to 10% for
the NE data. Since the kNN-R’s APC is almost controlled by the standard deviation, $\sigma$, of the noise component, we change $\sigma$ from 0.1 to 0.5 for normalized data distributions in $\mathcal{N}(0, 1)$. In comparison, Figure 9 shows the precision of kNN-R approach keeps steady around 56% for the uniform data and 44% for the NE data when we increase the standard deviation of noise. In addition, the increased APC does not affect the performance of query processing. With $E = 10$ for the k minimum range algorithm, it takes about 8 rounds and 0.8 seconds to finish 1000 queries.

6 Related work

We propose to evaluate an outsourced service based on the CPEL criteria: data Confidentiality, query Privacy, Efficient query processing, and Low inhouse workload. We discuss some of the related work based on these criteria.

Distance-recoverable encryption is the most intuitive method for preserving the kNN relationship. However, this type of encryption is vulnerable to known plaintext-ciphertext attack, as discussed in [22, 14, 6]. Wong et al. [22] notices that comparing dot products instead of distances is sufficient to find kNN. However, this approach depends on the strong assumption that attackers cannot know plaintext-ciphertext query pairs, which is impossible in practice. Once the attacker knows one pair of plaintext-ciphertext query it is straightforward to reconstruct the key transformation matrix. In addition, the encrypted data cannot be indexed, which results in low server efficiency.

Hu et al. [10] addresses the query privacy problem and requires the authorized query users, the data owner, and the cloud to collaboratively process kNN queries. It makes implicit assumption that authorized query users do not collude with the cloud to breach the security. Collusion is possible when the curious cloud provider disguises as a client to submit queries. Furthermore, most computation depends on the query user’s local processing and multiple rounds of interaction with the cloud server. The cloud server only aids query processing, which doesn’t meet the principle of letting the cloud take most responsibility of
query processing. Papadopoulos et al. [20] also focuses on the query privacy problem, specifically, location privacy, with private information retrieval methods [7]. This approach does not consider protecting the confidentiality of data.

SpaceTwist [24] proposed a method to query kNN with the encrypted users’ positions for location privacy. But it didn’t keep the data encrypted when it comes to sensitive data. The Casper approach [19] considers both data confidentiality and query privacy. It cloaks each data point and query point with a rectangle. The point can be anywhere within the cloaking box. Therefore, the size of cloaking box determines the level of protection. An algorithm is designed to find the nearest neighbor of the cloaked query point from the cloaked data points. Because of cloaking the query result may contain irrelevant points. Depending on the size of the domain, an acceptable size of cloaking might be large for some applications. Our experiment shows that the query result precision drops dramatically with the slightly increased size of cloaking.

Yiu et al. [15] uses a hierarchical space division method to encode spatial data points. It partitions the data space into blocks. In each block linear transformations are applied to $x$-axis and $y$-axis, respectively. Note that this transformation does not change the order of the $x$ and $y$ axis values in each block. Consequently, the dimensional order is preserved for the entire domain. The security weaknesses of order preserving encryption have been thoroughly discussed in several places [2, 4].

7 Conclusion

We propose to study an outsourced service based on the CPEL criteria: data Confidentiality, query Privacy, Efficient query processing, and Low inhouse workload. With the CPEL criteria in mind, we develop the kNN-R approach for secure outsourced kNN query service. The kNN-R approach takes advantage of fast and secure RASP range query processing to implement kNN query processing. It can find high precision kNN results and also minimize the interactions between the cloud server and the inhouse client. High precision kNN
results and minimized interactions result in low inhouse workload. We have conducted a thorough security analysis on data confidentiality and query privacy. Compared to the related approaches, the kNN-R approach achieves a better balance over the CPEL criteria.
References


