International Journal of Engineering & Technology, 7 (4) (2018) 4984-4990

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.25519
Research paper

Security of encrypted database in the cloud

Yousif H. Sulaiman *

AL-Maaref University College, Iraq, AL-Ramadi city
**Corresponding author E-mail:*

Abstract

Existing encryption techniques ensure or safeness, either effectiveness, but not both. most circuits even show the order of cipher tupelos , that allow opponents to accurately estimate the plain text values. This paper present -tree, hierarchic cipher index, which can be safely located in a cloud and effectively distorted. It is based on the arrangement, which was designed for cipher requests using the encryption method Asymmetrical scalar produce handling encryption (ASPE).

Keywords: *Query Encryption; Database Encryption; Cloud Computing; Hierarchical Index.*

1. Introduction

The article presents $\hat{R}$ -tree , hierarchic encoded index such can be reliably located in the cloud and effectively distorted. It is based on a machinery that was designed for encoded requests for a half-space range in R^d , applying asymmetrical encryption
Asymmetric scalar produce canning coding (ASPE).
Data holders be able to customize settings.
$\hat{R}$ -tree for achievement the required safety and efficiency
We also present performance evaluation experiments $\hat{R}$ -tree. Our effects show such queries $\hat{R}$ -tree performed in encoded data bank and show much little information than contesting techniques.

2. State of the problem

The term "cloud computing" relates to a wide diapason of outsourcing attendance for storing and computing [1]. This pattern is becoming more popular because clients keep practically unlimited exchequers, but above substantially, because they are released of the drag of controlling these exchequers. Therefore, outsourcing great databases plant oneself a good-studied theme.
But, this pattern has its expenses. Outsourced information many be encoded to maintain confidentiality and integrality, but encryption creates it difficult to execute requests. Usual encryption circuits, such as unit – ciphers, do not directly maintain the sorting of similes, quests and other operations required to process requests sans deprivation of confidentiality. Therefore, new encryption circuits [2, 3, 4, 5] to provide requests for encoded data.

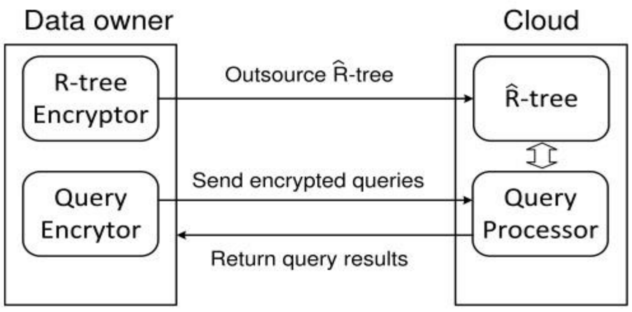

Fig. 1: Scheme of the Search Model Based on $\hat{R}$ -Tree.

Safeness and performance are significant thoughts while developing such coding schemes. several schemes [2, 4] reach performance by identifying the relative order of the encrypted data full stop, but the adversary can use the information to order the data accurately using order statistics [6]. Query schemes based on predicate-based encryption (PRE) [4], [7 - 9] provide reliable protection for encryption, but with high computational costs. Costs in these circuits increase greatly with a diapason of queries or the required accuracy.
Some leakage of order information is probably unavoidable, but the task is to minimize either leakage. For example, bouquetization schemes [3] create an exchange between the regulation of data about order and performance.

1) Formulation of the problem

An innovative method has been developed to perform encrypted requests for a half-space range in $\mathbf{R}^d$ over the full stops encoded since ASPE. This machinery can provide multi-faceted requests for cipher information. Applying this machinery, we provide $\hat{R}$ -tree, indexing circuit for cipher and outsourced data bank. $\hat{R}$ -tree utilizes ASPE to encode request ranges. The information itself may be cipher in every another method. $\hat{R}$ -tree is a hierarchic bouquetization circuit, yet, unlike the actual bouquetization circuit, encrypted indices $\hat{R}$ -tree be kept and requested bodily in the cloud, and not on the site of the information holder. $\hat{R}$ -trees allow us to more efficiently transfer data to data control.

2) Asymmetric encryption ASPE

In [10] was offered asymmetric encryption using a scalar product (ASPE) to perform kNN requests at encrypted data points in R^d. Encryption uses as a secret key $(d+1)\times(d+1)$ inverse matrix M. Data and requests are encrypted differently, which is reflected in our notation.

$Point\_Enc(P,M)\rightarrow\langle P\rangle$. It function takes a facts full stop. $P\in R^d$ and $(d+1)\times(d+1)$ crucial array M and displays the cipher text $\langle P\rangle$ of P. Initially makes a full stop $P_+\in R^{d+1}$, either that $P_+=(P^T\,|\,(-0.5\|P\|^2))^T$ - Euclidean norm P. Encryption p equals $P=M^TP_+$.

$Query\_Enc\left(Q,M^{-1}\right)\rightarrow[Q]$. It function takes a request full stop. $Q\in R^d$ and M^{-1}, inverse key matrix M. She gives out $[Q]$, cipher text Q. First he creates a point $Q_+\in R^{d+1}$, either, such $P_+=(P^T\,|\,(-0.5\|P\|^2))^T$, where r - random positive number. Encrypted dot $[Q]=M^{-1}Q_+$.

$Dist\_Comp(\langle P\rangle,\langle P'\rangle,[Q])\rightarrow\{0,1\}$. It function takes to encrypted facts points $\langle P\rangle,\langle P'\rangle$, encrypted request point $[Q]$ and returns 1 if , P closer to Q, than P'. It outputs a logical value. $(\langle P\rangle-\langle P'\rangle)\cdot[Q]>0$. now,

$$\begin{aligned}(\langle P\rangle-\langle P'\rangle)\cdot[Q]&=(\langle P\rangle-\langle P'\rangle)^T[Q]=\\&=(M^T(P_+-P'_+))^TM^{-1}Q_+=\\&=(P_+-P'_+)^TQ_+=\\&=\left(P-P'\right)^T\left(rQ\right)+r(-0.5\|P\|^2+0.5\|P'\|^2)=\\&0.5r(\|P'-Q\|^2-\|P-Q\|^2),\end{aligned}$$

Wherever $\|P-Q\|$ - Euclidean spacing betwixt P and Q. This term is positive if P nearer to Q, wherewith P'. The kNN request identify k immediate full stops by simile the spacing from the full stops of request Q to every facts full stop P.

2.1. Half-space range requests

Half-space range requests (hRQ) are a radical issue in calculative geometry, because either shape of search for an algebraic diapason may be transformed in it .If $\mathbf{a}\in\mathbf{R}^d$, $\mathbf{a}\neq 0$ and $b\in\mathbf{R}$, then the hyperplane H determined by the kit $\mathbf{x}\in R^d$ so, that $\mathbf{a}^T\mathbf{x}=b$. H breaks down $\mathbf{R}^d$ on the inner half space $\mathrm{H}^{\leq}$, corresponding $\mathbf{a}^T\mathbf{x}\leq b$ and outer half space $\mathrm{H}^{>}$, corresponding $\mathbf{a}^T\mathbf{x}>b$. Each $\mathrm{S}\in\mathbf{R}^d$ is broken H into two dislocate subkits $\mathrm{S}_{\mathrm{H}}^{\leq}=\mathrm{S}\cap\mathrm{H}^{\leq}$ and $\mathrm{S}_{\mathrm{H}}^{>}=\mathrm{S}\cap\mathrm{H}^{>}$.

Given the many points $\mathrm{S}=\{P_1,P_2,...,P_n\}$ and hyperplanes H in $\mathbf{R}^d$, request for a range of half range requests $\mathrm{S}_{\mathrm{H}}^{\leq}$.

2.2. Order statistics

Order statistics is an significant instrument in non-parametric statistics. [6]. Let be $X_1,X_2,...,X_n$ – random variables with density and distribution functions $f(x)$ and $F(x)$ respectively. Let be X_i sorted to get $X_{(1)}\leq X_{(2)}\leq...\leq X_{(n)}$. now X_k called statistics k -order. It can be shown that the density function X_k is given by the formula:

$$f_{X_{(k)}(x)}=\binom{n}{1}\binom{n-1}{k-1}f(x)\left[F(x)\right]^{k-1}\left[1-F(x)\right]^{n-k} \qquad (1)$$

2.3. Overview

Now we will present our circuit and safeness pattern and give an overview R -tree. Our access , in contradistinction to [10], utilizes an index to acceleration requests . In addition, we separate the encryption of facts full stops of query and index data.

2.3.1. Scheme pattern

Our pattern identifies two objects: the facts holder and the cloud service producer (Fig. 2). The fact holder accommodates the cipher facts and the appropriate index. R -tree in the cloud that ensures the infrastructure for calculation and keeping . The fact holder makes and transmits cipher requests for diapasons of concern to the cloud. The cloud executes cipher requested in the index R -tree and give back the request effects to the fact holder.

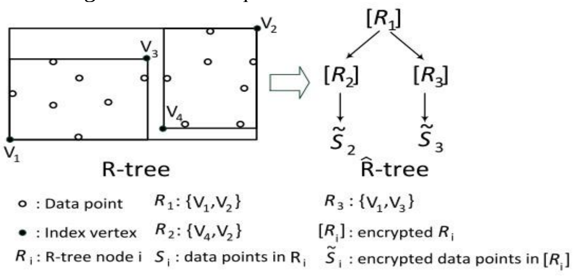

Fig. 2: R -Tree and $\hat{R}$ -Tree.

The facts holder makes $\hat{R}$ -tree, initially creating a orderly R -tree for adjusted kit of points $\mathrm{S}\in\mathbf{R}^d$. MBR diapasons are cipher utilizing ASPE to getting $\hat{R}$ -tree. Parent-child relationship $\hat{R}$ -tree not encrypted. Although the MBR ranges R -tree cipher utilizing ASPE to aid diapason requests, facts points in S can be cipher whatever of each other using other encoding circuits, such as ciphers-unit. d - measured diapason may be determined by its two extreme tops. R -tree in Fig. 2 comprise knots with MBR $\mathrm{R}_1=(V_1,V_2)$, $\mathrm{R}_2=(V_1,V_2)$ and $\mathrm{R}_3=(V_1,V_2)$. The data set comprised in R_2 and R_3, represents S_2 and S_3 accordingly.

MBR diapasons in $\hat{R}$ -tree encoded by using ASPE to every extreme peak utilized to determine the diapason . so demonstrated in fig. 2, conforming to $\hat{R}$ -tree comprises three knots $[R_1]=([V_1],[V_2])$, $[R_2]=([V_4],[V_2])$, $[R_3]=([V_1],[V_3])$.

The facts points within every MBR sheet are encoded applying the usual coding circuit . encoded releases S_2 and S_3 are $\tilde{\mathrm{S}}_2$ and $\tilde{\mathrm{S}}_3$.

The data holder makes encoded diapason requests and transmits their to the eddy.

Cloud is looking for $\hat{R}$ -tree, performing crossing reactions in stages as usual R -tree, and passes to the child nodes if and only if the bounding rectangle of the node intersects the range of requests. The cloud thus receives all the layers that intersect the range of requests, and it returns the encoded data points in these sheets to the data holder. The cloud cannot request the data points themselves, so far as they are encoded severally. $\hat{R}$ -tree can enter spurious positives in the request effects, yet defends the order information within every sheet MBR a sensible compromise. Exact request patterns [4, 8], whether comeback an exact kit of encoded tuples in a diapason of requests , can eventually leaks. Given sufficient diapason request effects, the opponent can restore the ordering of the tuples from the associations and crossing of these clusters of results.

2.3.2. Safeness pattern

We accept an "honest, but curious" pattern for our opponent, the eddy server. Its purpose is to study open texts for encoded facts. He can see some lore of the outsourced data kit and attempt to utilize this lore to get the point senses in the data kit. Else, he

scrupulously follows the protocol determined by the owner of the data and returns the correct results of the query.
ASPE, whether we utilize to encode diapasons of indexes and requests, is protected from famous plain text assails [10]. However, the enemy can create above compound assails. For example, it may get some data of the ordering of encoded data senses when handling requests. An opponent can see the cleartext allocation of every data full stops and the values of some facts points. It will attempt to evaluation the assails of another facts points applying either information.
We begin with the assumption that the opponent knows the order of the encrypted data points. Some encoding circuits, such as [2, 4], clearly show this ordering. In another events, it can be probable to derive this ordering above time of requests. It is as will frequently probable to get allocation of data assails or of publicly available founts, or by studying another accessible and parallel data gets.
Utilization this lore of allocations and ordering, an opponent may utilize method statistics to evaluate plaintext assails for encoded tuples. For one-dimensional data, let's tell that the opponent is studying the assails of the plaintext m data points $y_{i1} < y_{i2} < ... < y_{im}$. It utilizes these points so endpoints for deriving ranges. $m-1$ $[y_{i1}, y_{i2}],...,[y_{im-1}, y_{im}]$. He see the order of the encoded tuples in every diapason and may anon get better estimates of the plain text values of the encrypted tuples in each diapason applying order statistics. For multidimensional data, the opponent may execute the same assail in order to rather evaluate the senses of the encoded tuples for every measurement.
$\hat{R}$ -trees do not disclose the complete order of data points, but contain data about the leakage of information about the ordering of sheet MBRs. We will examine the performances of the assaults described up ward on $\hat{R}$ -trees.

2.3.3. Half space range request for encrypted data

Requests in [10] inquire whether encoded data points in $\mathbf{R}^d$ nearest to the present encoded request point. Then technique turns every data point to $\mathbf{R}^d$ to point to $\mathbf{R}^{d+1}$, further measurement incretion the distance of the point from the descent. Even so, request points are not demanded to transmit such interval information. Our treatment to encoded half-space diapason requests (EhQ) is binary to this way and should test whether of the two request points is nearer to the top in the MBR $\hat{R}$ -trees. Consequently, the request points are included with the interval information in our circuit , whereas the points conforming to the peaks of the MBR are not.
We build requests for half-space, as in Fig. 3. For hyperplane H and appropriate half spaces $\mathrm{H}^{\leq}$ and $\mathrm{H}^{>}$ we choose the pivot points $\omega^{\leq} \in \mathrm{H}^{\leq}$ and $\omega^{>} \in \mathrm{H}^{>}$, equidistant from H such that cut $(\omega^{>}, \omega^{\leq})$ orthogonal H. Each point on H now equidistant from $\omega^{\leq}$ and $\omega^{>}$, but points in $\mathrm{H}^{\leq}$ nearer $\omega^{\leq}$, a point in $\mathrm{H}^{>}$ nearer $\omega^{>}$. We may test if the percent point V in $\mathrm{H}^{\leq}$ or $\mathrm{H}^{>}$, testing if V nearer to $\omega^{\leq}$ or $\omega^{>}$, As in ASPE.

2.3.4. Index lookups as the intersection of hyper-intersections

Search $\hat{R}$ -trees requires to determine if hyper crosses a rectangle d - dimensional query hyper rect index in node $\hat{R}$ -trees. We create the regular aadmission that the coordinate axis is orthogonal and that every front of a hyper rectangle is orthogonal to some axis.

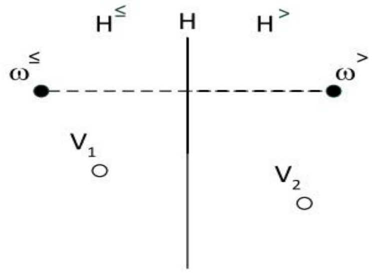

Fig. 3: Query the Half-Space Range.

Our method is based on the observation that d - dimensional hyper query rectangle Q can be defined as a space enclosed by hyperplanes $\mathrm{H}_1, \mathrm{H}_2, ..., \mathrm{H}_{2d}$, certain of it $2d$ facets. We accept the agreement that the scope of the request is defined in $\mathrm{H}_i^{\leq}$ for each i. That is, the hyperplanes are so refined that the points of interest $\mathbf{x}$ satisfy the condition $\mathbf{a}_i^T \mathbf{x} \leq b_i$. Under these conditions, we will have $Q = H_1^{\leq} \cap H_2^{\leq} \cap \ldots \cap H_{2d}^{\leq}$. Fig. 4 shows a two-dimensional request rectangle deermined by four queries in a half-space range, or eight control points. Half spaces $\mathrm{H}_i^{\leq}$ and $\mathrm{H}_i^{>}$ defined by two reference points $\omega_i^{\leq}$ and $\omega_i^{>}$. Choose $\omega_i^{\leq}$ randomly in $\mathrm{H}_i^{\leq} - \mathrm{H}_i$. $\omega_i^{>}$ will be its rerepulse in the hyperplane H_i. We itemize every index of the hyper rectangle. $\mathrm{R} \subseteq \mathbf{R}^d$ in the node $\hat{R}$ -trees their two peaks , as demonstrated in Fig. 2
In fig. 4 demonstrates three events of two-dimensional rectangular intersections. It is clear that we cannot verify the intersection of the rectangles simply by checking whether one vertex is embedded in another. Vice, we should check which the peaks are in the corresponding half-spaces defined by the query fronts. Q .

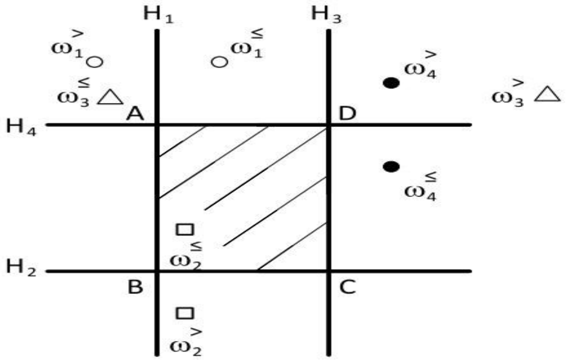

Fig. 4: Right Rectangle ABCD.

3.3. Our scheme

Our approach to the EhQ works as follows. To encode the request $Q = H_1^{\leq} \cap H_2^{\leq} \cap \ldots \cap H_{2d}^{\leq}$, we create anchors $\omega_i^{\leq}$ and $\omega_i^{>}$ for every hyperplane H_i. We then generate an encoded discriminator. Δ_{H_i} for every H_i. Applying Δ_{H_i}, we can define below encoding whether this cipher point is located V в $\mathrm{H}_i^{\leq}$ or in $\mathrm{H}_i^{>}$.

3.3.1. Encryption range peak and request algorithms

Our technique apply the next algorithms.
$Enc\_Vertex(V, M) \rightarrow [V]$. The data holder apply this algorithm to encode the peak . V MBR node $\hat{R}$ -trees, using your secret key M, inverse matrix $(d+1) \times (d+1)$.

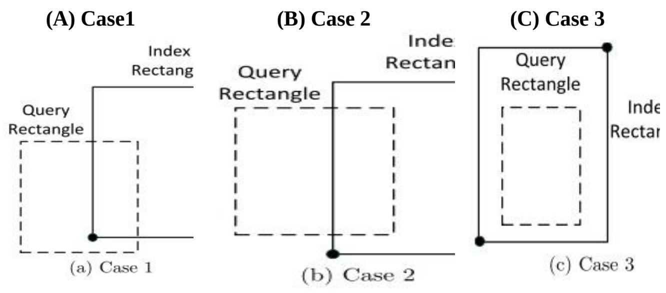

Fig. 5: Three Events of Crossing of Rectangles.

For a given vertex $V = (v_1, v_2, ..., v_d)^T$ the algorithm premier adds extra measurement to make $V_+ = (V^T | 1)^T$. Peak V encode to $[V] = r_1 M^{-1} V_+$, where r_1 – random positive number.

$Gen\_Anchor(\mathrm{H}) \rightarrow (\omega_i^{\leq}, \omega_i^{>})$. This algorithm take a given hyperplane. H, defined by arguments a and b and prints pivot points $\omega_i^{\leq}$ and $\omega_i^{>}$, recumbent in $\mathrm{H}^{\leq}$ and $\mathrm{H}^{>}$ accordingly. He randomly chooses a point. $\omega_i^{\leq} \in \mathrm{H}^{\leq} - \mathrm{H}$ and calculates $\omega^{>}$ as its reflection in H in the following way. If a $\omega^{\leq}$ and $\omega^{>}$ – vectors representing $\omega^{\leq}$ and $\omega^{>}$ accordingly, we require their vector difference $\omega^{\leq} - \omega^{>}$ was orthogonal to H. of lineal algebra we see that the vector $\mathbf{a}$ orthogonal to H. Let be $\mathbf{a}^T \omega^{\leq} - b = \delta$. We have $\mathbf{a}^T \omega^{>} - b = -\delta$, $\mathbf{a}^T(\omega^{\leq} - \omega^{>}) = 2\delta$. Because $\mathbf{a}$ and $\omega^{\leq}$ known we can get $\omega^{>} = \omega^{\leq} - \frac{2\delta}{\|\mathbf{a}\|^2}\mathbf{a}$.

$Gen\_Discr(\omega^{\leq}, \omega^{>}) \rightarrow \Delta_H$. This algorithm takes pivot points $\omega^{\leq} = (\omega_1^{\leq}, \omega_2^{\leq}, ..., \omega_d^{\leq})^T$ and $\omega^{>} = (\omega_1^{>}, \omega_2^{>}, ..., \omega_d^{>})^T$, corresponding hyperplanes H, and displays the discriminator Δ_{H}. Former, he adds distance data to reference points to get $\omega_+^{\leq} = ((\omega^{\leq})^T | (-0.5\|\omega^{\leq}\|^2))^T$ and $\omega_+^{>} = ((\omega^{>})^T | (-0.5\|\omega^{>}\|^2))^T$. Further $\omega_+^{\leq}$ and $\omega_+^{>}$ encrypted using M as $\langle\omega^{\leq}\rangle = M^T \omega_+^{\leq}$ and $\langle\omega^{>}\rangle = M^T \omega_+^{>}$. Eventually, the algorithm chooses a casual positive sense. r_2 and makes an encoded discriminator $\Delta_{\mathrm{H}} = r_2(\langle\omega^{\leq}\rangle - \langle\omega^{>}\rangle)$.

3.3.2. Half-space range requests in encrypted vertices of the MBR

The cloud performs an encrypted half-space range request at encoded MBR peaks applying the incoming algorithms.

$Halfspace\_Qry([\mathrm{V}], \Delta_{\mathrm{H}}) \rightarrow \mathrm{V}_{\mathrm{H}}^{\leq}$. This office takes a kit of encode MBR peaks. $[\mathrm{V}]$ and hyper plane discriminator Δ_{H} and prints a lot $\mathrm{V}_{\mathrm{H}}^{\leq} = \mathrm{V} \cap \mathrm{H}^{\leq}$.

It works by vocation the incoming office to check every point in $[\mathrm{V}]$.

$In\_Halfspace([\mathrm{V}], \Delta_H) \rightarrow \{0,1\}$. The office takes an encoded point. $[\mathrm{V}]$, discriminator Δ_{H} and gives a bit specify which $\mathrm{V} \in \mathrm{H}^{\leq}$ calculate $\Delta_H \cdot [\mathrm{V}]$. Because

$$\begin{aligned}
&\Delta_H \cdot [\mathrm{V}] = r_1 r_2(\langle\omega^{\leq}\rangle - \langle\omega^{>}\rangle) \cdot [\mathrm{V}] = \\
&= r_1 r_2((M^T \omega_+^{\leq}) - (M^T \omega_+^{>})^T M^{-1} \mathrm{V}_+ = \\
&= r_1 r_2(\omega_+^{\leq} - \omega_+^{>})^T \mathrm{V}_+ = \\
&= r_1 r_2(\|\omega_+^{\leq} - \mathrm{V}\|^2 - \|\omega_+^{>} - \mathrm{V}\|^2)^T \mathrm{V},
\end{aligned}$$

$\Delta_H \cdot [\mathrm{V}] \geq 0$, if V is in $\mathrm{H}^{\leq}$. Function exits 1, if V is in $\mathrm{H}^{\leq}$ and 0 other.

3.3.3. Hyper rectangle intersection

We display how to define the intersection between encoded d-spatial request and hyper rectangles of the index based on queries in the range of a half-space. We require that each surface of the hyper rectangle be orthogonal to the coordinate axis. That is,every front is a hyperplane $\mathrm{H}_i = (x_1, ..., x_{i-1}, c_i, x_{i+1}, ..., x_d)$, where c_i – constant ,but x_i not limited. This restriction is necessary, so far as crossing checks applying half-space requests can not operate on common polyhedra, so we shall spot.

Hyperrect Index $\mathrm{R} \subset \mathbf{R}^d$ now completely determined by its extreme vertices $V_\perp, V_\cdot \in R^d$, defined as follows. If a $V = (v_1, v_2, ..., v_d)$ represents the top R, then we define $V_\perp = (min\{v_1\}, ..., min\{v_d\})$, $V_\cdot = (max\{v_1\}, ..., max\{v_d\})$, where min and max taken on all heights V of R.

Nevertheless, the hyperdirect of a query is defined in fee of half-spaces defined by its fronts, since $\mathrm{Q} = \mathrm{H}_1^{\leq} \cap \mathrm{H}_2^{\leq} \cap \ldots \cap \mathrm{H}_{2d}^{\leq}$.

Hyper The index rectangle is decrypted as follows. $Enc\_Index(\mathrm{R}, M) \rightarrow [\mathrm{R}]$. Given the hyper rectangle index $\mathrm{R} = (V_\perp, V_\cdot)$ and clef array M this algorithm displays encryption R as $[R] = ([V_\perp], [V_\cdot])$, $[V_\perp] = Enc\_Vertex(V_\perp, M)$ and $[V_\cdot] = Enc\_Vertex(V_\cdot, M)$.

$Enc\_Query(Q, M) \rightarrow Q$. This algorithm takes a key matrix and a request area. Q, given as the crossing of half spaces $\mathrm{H}_1^{\leq} \cap \mathrm{H}_2^{\leq} \cap \ldots \cap \mathrm{H}_{2d}^{\leq}$. For each H_i he first calls $Gen\_Anchor(\mathrm{H}_i)$, To obtain $\omega_i^{\leq}$ and $\omega_i^{>}$. Then he gets Δ_{H_i}, causing $Gen\_Discr\ (\omega_i^{\leq}, \omega_i^{>})$. It returns the area of the encrypted request. $\langle\mathrm{Q}\rangle = (\Delta_{\mathrm{H}_1}, \Delta_{\mathrm{H}_2}, ..., \Delta_{\mathrm{H}_{2d}})$.

$Xsect\_Index([\mathrm{R}], \mathrm{Q}) \rightarrow \{0,1\}$. This function accepts query and index hyper rectangles, either encoded. It exits a Boolean value specify which the hyper rectangles cross. when either $V_\perp$ and $V_\cdot$ defined as lying outside $\mathrm{H}_i^{\leq}$ for some H_i, algorithm returns 0 and 1 else (spot Algorithm 1).

Algorithm 1. $Xsect\_Index$

```
input: $[R] = ([V_\perp], [V_\cdot])$, $\langle \mathrm{Q} \rangle = (\Delta_{\mathrm{H}_1}, ..., \Delta_{\mathrm{H}_{2d}})$
output: $\{0,1\}$
  1) for each $\Delta_{\mathrm{H}_1} \in \langle \mathrm{Q} \rangle$ do
  2)     if not $In\_Halfspace([\mathrm{V}_\perp], \Delta_{\mathrm{H}_1})$, and not $In\_Halfspace([\mathrm{V}_\cdot], \Delta_{\mathrm{H}_1})$
then
  3)         return 0
  4)     end
  5) end
  6) return 1
```

3.3.4. Versatile request region

Our circuit may cultivate voluntary protuberant versatile request areas, yet it can bring in pseudo positives. The two events demonstrated in fig. 6, can not be outstand. (a) No crossing (b) crossing

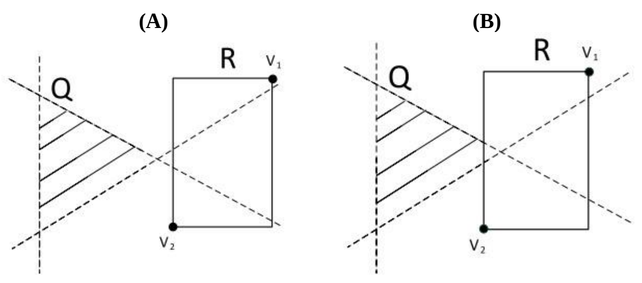

Fig. 6: Can Not Be Outstanding. (A) No Crossing (B) Crossing.

Fig. 6. Half-space requests applying a multi-faceted area Both cevents comeback the same results if we e half-space requests for the peaks of the hyper rectangle of the index using $H_i^{\leq}$, determinant the scope of lateen queries. nevertheless, our circuit is secure for prominent multifaceted request areas, so far as it does not enter pseudo negatives in accordance with proposition 1.

Proposition 1. $Xsect\_Index([R],\langle Q\rangle)$ gives out 0, if the convex region of the versatile request Q and index diapason R do not cross.

Evidence. R certain by the formula $V_{\perp} = (min\{v_1\},...,min\{v_d\})$ and $V_{\cdot} = (max\{v_1\},...,max\{v_d\})$, its extreme peaks. If in $In\_Halfspace([V_{\perp}],\Delta_{H_1})$ and $In\_Halfspace([V_{\cdot}],\Delta_{H_1})$ output 0 for floor space query $\Delta_{H_1} \in \langle Q\rangle$, either $V_{\perp}$ neither $V_{\cdot}$ is not in $H^{\leq}$. Because $V_{\perp}$ and $V_{\cdot}$ keep, accordingly, the littlest and biggest projections together the axis i, then neither 2^d - vertices in R cannot be in $H^{\leq}$. It's clear that Q and R do not cross.

3.3.5. Design and request $\hat{R}$ -trees

Tree $\hat{R}$ can be considered as R-tree, whose MBR is encrypted, but the relationship between parents and children is not. Itsinformation points are encoded aside. Thes encoded information points, encoded indexes $\hat{R}$ -trees and procreator-child relationships are located in the cloud. We check the overlapbetwixt encoded request diapasons and encoded MBR trees applying the new encoded Half space request (EhQ) arrangement.

Our EhQ treatment is safe and effective and can be utilized to quest for encrypted another compound data textures , such as BNL trees and trees. We decided to basis our index $\hat{R}$ -tree на R -tree, so far as the family R -trees has a lower information escape than bunching circuits, k - tools, BNL-tree, kD - a tree, etc., so demonstrated in [3].

$\hat{R}$ -trees built as in algorithm 2. Let S and $\tilde{S}$ denote encrypted versions of the data set. For an MBR sheet, have SR designate the data points that fall in R , but $\tilde{S}_R$ denote encrypted texts S_R . Let be $\mathbf{T}$ and $\hat{\mathbf{T}}$ note R -tree and the conforming $\hat{R}$ -tree accordingly. Let be $\mathbf{PC}$ denotes a kit of parental -child relations in $\mathbf{T}$.

The incoming office, developed in algorithm 3, prize all the leaves of a tree. that cross the specified diapason request.

$\hat{\mathbf{R}}-tree\_Qry(Q,\hat{\mathbf{T}}) \rightarrow \mathbf{L}$ This function accepts a request as input. $\langle Q\rangle$ encrypted range, $\hat{R}$ -tree $\mathbf{T}$. Her exit - kit of encoded leaves $\mathbf{L}$, whose MBR cross $\langle Q\rangle$.

3.4. Safeness analysis

Our analysis display that outsourcing circuits that eliminate simple tuple encryption cannot provide reliable confidentiality ensures. Such order information frequently allows an opponent to accurately assessment the senses of the encode tuples. We start by analyzing the safety of the circuit utilized to encrypt the hypersurfaces of the index and request in $\hat{R}$ -trees. Then we collate the confidentiality ensures given that by our circuit with those given that by contesting circuits, particularly when the opponent was able to detect fractional data around the senses of the encoded tuples.

Algorithm 2. Construction $\hat{R}$ -tree
input: $\mathbf{T}$, M
output: $\hat{\mathbf{T}}$

```
1)  $\hat{\mathbf{T}} = \varnothing$
2)  $\mathbf{PC} = \varnothing$  $stack = \varnothing$
3)  $node = \mathbf{T}$.root
4)  if node $\neq$ NULL then
5)  $stack.Push(node)$
6)  end
7)  else
8)  return $\varnothing$
9)  end
10) while $stack \neq \varnothing$ do
$node = stack.Pop()$
// $node$.R $denotes$ $node's$ MBR
11) [$[R] = Enc\_Index(node.R, M)$ )
12) if node has children then
13) for each child do
14) Save the parent-child relationship to $\mathbf{PC}$
15) end
16) end
17) if node is not a leaf then
18) for each child do
19) $stack.Push(child)$
20) end
21) // Generate a node for $\hat{\mathbf{T}}$
22) $onode = \{[R]\}$
23) else
24) Encrypt data points in $S_R$ to obtain $\tilde{S}_R$
25) $onode = \{[R], \tilde{S}_R\}$
26) end
27) Add $onode$ to $\hat{\mathbf{T}}$
28) end
29) Add $\mathbf{PC}$ to $\hat{\mathbf{T}}$
30) return $\hat{\mathbf{T}}$
```

Algorithm 3. $\hat{R}$ -tree Qry
entrance: $\langle Q\rangle$, $\hat{\mathbf{T}}$

```
1)  $\mathbf{L} = \varnothing$
2)  $stack = \varnothing$
3)  $node = \hat{\mathbf{T}}.root$
4)  if $Xsect\_Index(node.[R],\langle Q\rangle)$ then
5)  $stack.Push(node)$
6)  end
7)  else
8)  return $\varnothing$
9)  end
10) while $stack \neq \varnothing$ do
11) $node = stack.Pop()$ )
12) if $node$ $is$ $a$ $leaf$ then
13) $\mathbf{L} = \mathbf{L} \cup node$
14) end
15) else
16) for each node's child do
17) if $Xsect\_Index(child.[R],\langle Q\rangle)$ then
18) $stack.Push(child)$
19) end
20) end
21) end
22) end
23) return $\mathbf{L}$
```

Table 1: Encrypted Database Schemas. N Tale of Tuples C - the Extent of the Bouquet. (Maximum Burden log$N$ [2], [4] May Merely Be Reached for One-Dimensional Data)

circuit	request Overheads	Indicate course?
Bucketization [3]	High: $O(N/C)$	No.
course canning [2]	Low: log$N$	Yes.
Predicate encoding[4]	Low: log$N$	Yes.

3.4.1. Encryption security

We encode the subject-heading and request hyperdiacles with ASPE, whether was proven to be safe with plain text assaults in [10]. Our circuit saves the security attributes of ASPE, because it proceeds ASPE, yet does not change the main treatment to encoding in [10]. synthetic dimensions and casual asymmetric cleavage more operate in our circuit.

3.4.2. Comparison since contesting circuits

We show as order data can be used by an opponent to display cleartext senses applying course statistics. so shown in the tspreadsheet. 1, the current circuits give either performance or solitude defense, yet not either. So, the bouquetization circuit from [3] advocates the order data, yet perishes because of the tall request burden. It as well demands the holder of the data to control the bouquet indexes . In relief , [2], [4] permit efficacious requites , but display order data on encoded tuples. $\hat{R}$ -Tree can perform highly efficacious requests by cloaking the order of data points in every MBR.
For some causes , we will not conduct a itemized comparison of our circuit since the bouquetization circuit [3]. Former, [3] is not a genuine outsourcing circuit, so far as the index is stored on the data holder's site, and not in the cloud. It as well demands that all requests are executed by the data holder, and this means significant requirements. Finally, index quest accepts time. $O(N/C)$, that is lineal in the scale of the database if we hold the scale of the bouquet invariable. These overheads are superfluous ccompared to cotesting circuits.
Therefore, we collate the stability of our circuit with the ability of the circuits in [2], [4]. This is a fitting matching , so far as $\hat{R}$ -tree achieves the same way request intricacy so these schemas. We will show that our circuit has more rather stability, an edge that it has contrary each circuit such does not mask the ordering of the tuples.

3.4.3. Attack model

Let be A_O denotes an adversary in schemes that reveal information to streamline, and let $\mathrm{A}_{\hat{R}}$ denotes an opponent into our circuit. Leaf Index diapasons $\hat{R}$ -tree are limiting fields per clusters of encipher information points. upper layer knots are additional clusters of limiting blocks. $\mathrm{A}_{\hat{R}}$ cannot spot index diapasons at every of the nodes $\hat{R}$, because they are cipher using ASPE. Even so, $\mathrm{A}_{\hat{R}}$ able to study the ordering of every MBR sheets from a sufficient number of query results. $\mathrm{A}_{\hat{R}}$ can selectively take a subspace discriminator Δ_{H_i} from the received requests for the formation of new requests. But these requests can help him get an order of leafy MBR. The order of the data points within every sheet MBR is more safe.
purpose $\mathrm{A}_{\hat{R}}$ is to output the senses of the cipher data points allocable to the knot λ_j sheet. $\mathrm{A}_{\hat{R}}$ can examine the plain text senses from little cipher data points. Let's pretend that $\mathrm{A}_{\hat{R}}$ knows the lower and upper limits of the diapason in λ_j , and he too have the allocation from point senses.
For match our circuit since procedures such as [2], [4], which show the order of cipher points, we suppose such A_O too attempts to output the senses of the cipher data points to λ_j . A_O Have the relative order of every data points, the lower and upper limits of the diapason into λ_j and the allocation from point senses.

3.4.4. The optimal evaluation criterion of the enemy

The purpose of the enemy is to display the senses of the cipher tuples. For this end, it will utilize a statistic estimate, the performance of which should be calculated in terms of the fault it introduces. We will utilize the extensively utilized metric of the root mean square estimate (MSEE), too utilized in [3], that works so tracks. Per rusticity, regard the one-dimensional matter. It is frequently necessary to evaluate the sense of a casual alternate. Y , which itself is not available in terms of the function $g(X)$, available casual alternative X . In our matter Y is a plain text tuple. The opponent select a suitable casual alternate. X . MSEE is spotted so $\mathbf{E}[(Y-g(X))^2]$. The easiest choice for an opponent is $g(X)=c$, constant. We find the value c_{min} , which minimizes MSEE as follows. Beginning with $\underset{c}{min}\,\mathbf{E}[(Y-c)^2]=\underset{c}{min}\{\mathbf{E}[Y^2]-2c\cdot\mathbf{E}(Y)+c^2\}$,
We differentiate by and set to 0, we get $c_{min}=\mathbf{E}[Y]$. Minimum MSEE now $\mathbf{E}[(Y-\mathbf{E}[Y])^2=Var(Y)$. Therefore, the optimal estimate for Y is an $\mathbf{E}[Y]$, which reaches the minimum MSEE value $Var(Y)$.
Therefore, given the encrypted tuple $\tilde{y}_i$ open text y_i , taken from the allocation simulated by a casual alternate Y , better evaluation that an opponent may do to y_i , equal $\mathbf{E}[Y]$, reaching MSEE $Var(Y)$.

3.4.5. Installation attacks since and out of order information

present the continuous diapason $R=[y_s,y_e]$, containing $|R|$, encrypted tuples $(\tilde{y}_1,\tilde{y}_2,...,\tilde{y}_{|R|})$, оба adversary A_O и $\mathrm{A}_{\hat{R}}$ try to display plain text values y_s,y_e cipher tuples in R . We suppose both A_O and $\mathrm{A}_{\hat{R}}$ have the meanings of plaintext y_s,y_e of the two end points of the diapason R . Permit the causal alternate Y corresponds to the same allocation so the plaintext senses of every cipher tuples that have the solidity function $f(y\}$. Besides, A_O knows how distribution $f(y\}$ open texts and the order of the encrypted sets $\tilde{y}_i$. $\mathrm{A}_{\hat{R}}$ have the allocation $f(y\}$, yet not the order of the cipher tuples $\tilde{y}_i$.

3.4.5.1 Attack $\mathrm{A}_{\hat{R}}$ (order unknown)

Let the random variable Y_R matches the same allocation so the plaintext senses of the cipher tuples in R . Applying allocation $f(y)$, $\mathrm{A}_{\hat{R}}$ discovers allocation $f_{Y_R}(y)$ for Y_R . In 3.4.4, we saw that the best estimate $\mathrm{A}_{\hat{R}}$ for any $\tilde{y}_i \in R$ - $\mathbf{E}[Y_R]$.

3.4.5.2. Attack A_O (order known)

A_O can do much better since he knows the order $\tilde{y}_i$. A_O first finds $f_{Y_R}(y)$. Permit the causal alternate $Y_{(k)|R}(y)$ represents the plaintext sense of the kth littlest set in the diapason R , having an allocation $f_{(k)|R}(y)$. A_O gets $f_{(k)|R}(y)$, using $f_{Y_R}(y)$ and equation (1). Let be $\tilde{y}_{(k)}$ denotes k smallest set in R . As in § 3.4.4, the best score is A_O for $\tilde{y}_{(k)}$ th open text $y_{(k)}$ - $\mathbf{E}[Y_{(k)|R}]$.

3.4.6. Metrics ε absolute estimation errors

Let $\mathrm{A}_{\hat{R}}$ and A_O evaluate the true value of the plaintext y_i for encrypted tuple $\tilde{y}_i \in R$, as $y_i^{\hat{R}}$ and y_i^{O} accordingly. We determine the stark assessment fault for $\mathrm{A}_{\hat{R}}$ as $\varepsilon_{y_i}^{\hat{R}}=|y_i-y_i^{\hat{R}}|$ and for

A_O - $\varepsilon^{O}_{y_i} = |y_i - y^{O}_i|$. If a $\tilde{y}_{(k)}$ - smallest set in R, define $\varepsilon^{\hat{R}}_{(k)} = |y_{(k)} - \mathbf{E}[Y_R]|$ и $\varepsilon^{O}_{(k)} = |y_{(k)} - \mathbf{E}[Y_{(k)R}]|$.

3. Conclusions

The paper presents $\hat{R}$ -tree - A hierarchic cipher subject-heading that may provide save and effective diapason requires on cipher data. $\hat{R}$ -tree hides the order of internal MBR files to defend data privity. Our speculation and experiential assay show that identifying order is hazardous for external data, and $\hat{R}$ -tree has more rather stability than circuits out of streamlining information security. too developed a system that implements $\hat{R}$ -tree, having good performance.

References

[1] *Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. Commun. ACM, 53(4):50–58, April 2010.* https://doi.org/10.1145/1721654.1721672.

[2] *Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. Order-preserving symmetric encryption. In EUROCRYPT, pages 224–241, 2009.* https://doi.org/10.1007/978-3-642-01001-9_13.

[3] *Bijit Hore, Sharad Mehrotra, Mustafa Canim, and Murat Kant-arcioglu. Secure multidimensional range queries over outsourced data. The VLDB Journal, 21:333–358, 2012.* https://doi.org/10.1007/s00778-011-0245-7.

[4] *Yanbin Lu. Privacy-preserving logarithmic-time search on encrypt-ed data in cloud. In NDSS, 2012.*

[5] *Man, Lung Yiu, Gabriel Ghinita, Christian S. Jensen, and Panos Kalnis. Outsourcing search services on private spatial data. In ICDE, P. 1140–1143, 2009.*

[6] *H.A. David and H.N. Nagaraja. Order Statistics, Third Edition. Wiley, New York, 2003.* https://doi.org/10.1002/0471722162.

[7] *Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption sup-porting disjunctions, polynomial equations, and inner products. In Proceed-ings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology, EUROCRYPT'08, P. 146–162, 2008.*

[8] *Ming Li, Shucheng Yu, Ning Cao, and Wenjing Lou. Authorized private keyword search over encrypted data in cloud computing. In ICDCS, P. 383–392, 2011.*

[9] *Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in en-cryption systems. In TCC, P. 457–473, 2009.*

[10] *Wai Kit Wong, David Wai-lok Cheung, Ben Kao, and Nikos Ma-moulis. Secure knn computation on encrypted databases. In Pro-ceedings of the 35th SIGMOD international conference on Man-agement of data, SIGMOD '09, P. 139– 152, 2009.*