Special Issue of International Journal of Advances in Applied Science and Engineering (IJAEAS)
ISSN (P): 2348-1811; ISSN (E): 2348-182X
Vol. 4, Issue 1,2, March 2017, 40-45

GEOMETRIC RANGE SEARCH ON ENCRYPTED SPATIAL DATA

SHAIK.GULSHAN SANHEERA, Mr. KARIMULLA

PG Scholar ,QCET ,Nellore,AP,India

ASSOCIATE PROFESSOR, QCET ,Nellore,AP,India

ABSTRACT— ***Geometric range search is a fundamental primitive for spatial data analysis in SQL and NoSQL databases.It has extensive applications in location-based services, computer aided design, and computational geometry. Due to the dramatic increase in data size, it is necessary for companies and organizations to outsource their spatial data sets to third-party cloud services (e.g., Amazon) in order to reduce storage and query processing costs, but, meanwhile, with the promise of no privacy leakage to the third party. Searchable encryption is a technique to perform meaningful queries on encrypted data without revealing privacy. However, geometric range search on spatial data has not been fully investigated nor supported by existing searchable encryption schemes. In this paper, we design a symmetric-key searchable encryption scheme that can support geometric range queries on encrypted spatial data. One of our major contributions is that our design is a general approach, which can support different types of geometric range queries. In other words, our design on encrypted data is independent from the shapes of geometric range queries. Moreover, we further extend our scheme with the additional use of tree structures to achieve search complexity that is faster than linear. We formally define and prove the security of our scheme with indistinguishability under selective chosen-plaintext attacks, and demonstrate the performance of our scheme with experiments in a real cloud platform (Amazon EC2).***

I.INTRODUCTION

GEOMETRIC range search [1], [2] is one of the most fundamental queries performed on spatial data, where data are represented as points while queries can be described as geometric objects, such as triangles, circles, rectangles. It is an indispensable function, which is included in most SQL and NoSQL databases. For instance, major database applications,such as MySQL, Oracle, PostgreSQL (with additional use of PostGIS) and MongoDB, all provide certain types of geometric range search. The purpose of geometric range search on a spatial dataset is to retrieve points that are inside a particular geometric range Geometric range search is a critical tool for spatial data analysis, and has wide applications in geometric information

II.LITERATURE SURVEY

1) Filtering search: A new approach to query-an

AUTHORS: B. Chazelle

We introduce a new technique for solving problems of the following form: preprocess a set of objects so that those satisfying a given property with respect to a query object can be listed very effectively. Among well-known problems to fall into this category we find range query, point enclosure, intersection, near-neighbor problems, etc. The approach which we take is very general and rests on a new concept called fitering search. We show on a number of examples how it can be used to improve the complexity of known algorithms and simplify their implementations as well. In particular, filtering search allows us to improve on the worst-case complexity of the best algorithms known so far for solving the problems mentioned above.

2) Geometric range searching and its relatives

AUTHORS: P. K. Agarwal and J. Erickson

A typical range searching problem has the following form: Preprocess a set S of points in $\mathbf{R}^d$ so that the points in S lying inside a query range can be reported or counted quickly. We survey the known techniques and data structures for range searching and describe their applications to other related searching problems.

Range searching arises in a wide range of applications, including geographic information systems, computer graphics, spatial databases, and

time-series databases. Furthermore, a variety of geometric problems can be formulated as a range-searching problem.

3) Location privacy via private proximity testing

AUTHORS: A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh We study privacy-preserving tests for proximity: Alice can test if she is close to Bob without either party revealing any other information about their location. We describe several secure protocols that support private proximity testing at various levels of granularity. We study the use of "location tags" generated from the physical environment in order to strengthen the security of proximity testing. We implemented our system on the Android platform and report on its effectiveness. Our system uses a social network (Facebook) to manage user public keys.

4) Efficient reachability query evaluation in large spatiotemporal contact datasets

AUTHORS: H. Shirani-Mehr, F. Banaei-Kashani, and C. Shahabi

With the advent of reliable positioning technologies and prevalence of location-based services, it is now feasible to accurately study the propagation of items such as infectious viruses, sensitive information pieces, and malwares through a population of moving objects, e.g., individuals, mobile devices, and vehicles. In such application scenarios, an item passes between two objects when the objects are sufficiently close (i.e., when they are, so-called, *in contact*), and hence once an item is initiated, it can penetrate the object population through the evolving network of contacts among objects, termed *contact network*. In this paper, for the first time we define and study reachability queries in large (i.e., disk-resident) contact datasets which record the movement of a (potentially large) set of objects moving in a spatial environment over an extended time period. A reachability query verifies whether two objects are "reachable" through the evolving contact network represented by such contact datasets. We propose two contact-dataset indexes that enable efficient evaluation of such queries despite the potentially humongous size of the contact datasets. With the first index, termed *ReachGrid*, at the query time only a small necessary portion of the contact network which is required for reachability evaluation is constructed and traversed. With the second approach, termed *ReachGraph*, we precompute reachability at different scales and leverage these precalculations at the query time for efficient query processing. We optimize the placement of both indexes on disk to enable efficient index traversal during query processing. We study the pros and cons of our proposed approaches by performing extensive experiments with both real and synthetic data. Based on our experimental results, our proposed approaches outperform existing reachability query processing techniques in contact networks by 76% on average.

5) Computational Geometry: Algorithms and Applications

AUTHORS: M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars

This well-accepted introduction to computational geometry is a textbook for high-level undergraduate and low-level graduate courses. The focus is on algorithms and hence the book is well suited for students in computer science and engineering. Motivation is provided from the application areas: all solutions and techniques from computational geometry are related to particular applications in robotics, graphics, CAD/CAM, and geographic information systems. For students this motivation will be especially welcome. Modern insights in computational geometry are used to provide solutions that are both efficient and easy to understand and implement. All the basic techniques and topics from computational geometry, as well as several more advanced topics, are covered. The book is largely self-contained and can be used for self-study by anyone with a basic background in algorithms. In this third edition, besides revisions to the second edition, new sections discussing Voronoi diagrams of line segments, farthest-point Voronoi diagrams, and realistic input models have been added.

III. EXISTING SYSTEM

- While most of the searchable encryption schemes focus on common SQL queries, such as keyword queries and Boolean queries, few studies have specifically investigated geometric range search over encrypted spatial data.
- Wang et al. proposed a novel scheme to specifically perform *circular range* queries on encrypted data by leveraging a set of *concentric circles*.
- Some previous searchable encryptions handling order comparisons can essentially manage axis parallel rectangular range search on encrypted spatial data.
- Similarly, Order-Preserving Encryption, which has weaker privacy guarantee than searchable encryption, is also able to perform axis-parallel rectangular range search with trivial extensions.
- Ghinita and Rughinis particularly leveraged certain Functional Encryption with hierarchical encoding to efficiently operate axis-parallel rectangular range

search on encrypted spatial data in the application of mobile users monitoring.

3.1 DISADVANTAGES OF EXISTING SYSTEM

- Most of the searchable encryption schemes focus on common SQL queries, such as keyword queries and Boolean queries, few studies have specifically investigated geometric range search over encrypted spatial data.
- Inevitably introduces obstacles in terms of search functionalities over encrypted data.
- None of these previous works have particularly studied geometric range queries which are expressed as *non-axis-parallel rectangles* or *triangles*.
- More importantly, there still lacks a *general* approach, which can flexibly and securely support different types of geometric range queries over encrypted spatial data regardless of their specific geometric shapes.

IV. PROPOSED SYSTEM

- In this paper, we propose a *symmetric-key probabilistic* Geometric Range Searchable Encryption. With our scheme, a *semi-honest* (i.e., *honest-but-curious*) cloud server can verify whether a point is inside a geometric range over encrypted spatial datasets. Informally, except learning the necessary Boolean search result (i.e., *inside or outside*) of a geometric range search, the semi-honest cloud server is not able to reveal any private information about data or queries.
- Our main contributions are summarized as follows:
- We present a symmetric-key probabilistic Geometric Range Searchable Encryption, and formally define and prove its security with indistinguishability under Selective Chosen-Plaintext Attacks (IND-SCPA).
- In addition, our search process is *non-interactive* on encrypted data. In terms of search complexity, our baseline scheme incurs linear complexity (with regard to the number of data records), and its advanced version realizes faster than-linear search by integrating with tree structures.
- Our design is a general approach, which can securely support different types of geometric range queries on encrypted spatial data regardless of their geometric shapes. Furthermore, our design is not only suitable for geometric range queries, but also compatible with other regular types of geometric queries, such as *intersection queries* and *point enclosure queries*, over encrypted spatial data.

4.1.ADVANTAGES OF PROPOSED SYSTEM:

- The security of our scheme is formally defined and analyzed with indistinguishability under Selective Chosen-Plaintext Attacks.
- Our design has great potential to be used and implemented in wide applications, such as Location-Based Services and spatial databases, where the use of sensitive spatial data with a requirement of strong privacy guarantee is needed.

V.ALGORITHM:

1) Building a BC Decision Tree:

When the cutting of a prefix plane according to rule boundaries is performed, both the starting and the ending boundaries of each rule can be used for cutting, but cutting by either is sufficient since decision tree algorithms generally search for a subspace in which an input packet belongs and the headers of the given input are compared for entire fields to the

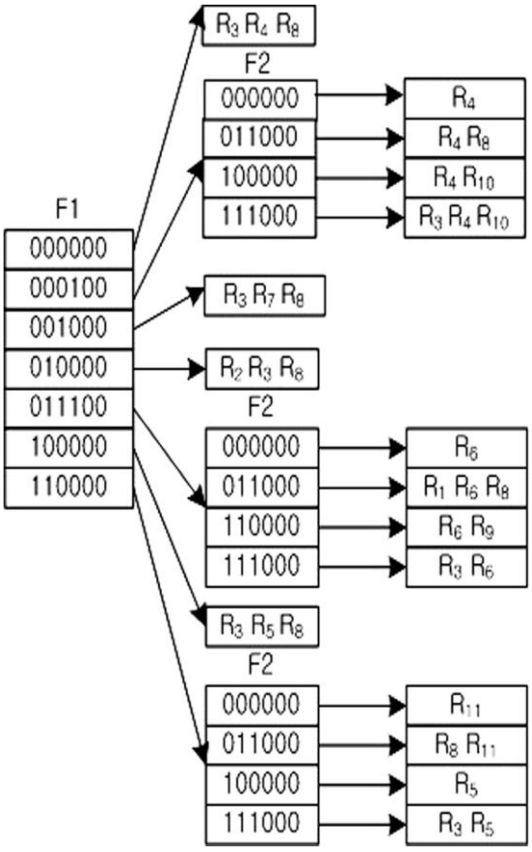


Fig: 3.2 Decision tree of the boundary cutting algorithm

Rules belonging to the subspace (represented by a leaf node of the decision tree). For example, regarding the rules shown in Fig. 1, by padding zeros to reach the maximum length (assuming six in this case), the start boundaries in the field of the rules can be derived as 000000, 000100, 001000, 010000, 011100, 100000, and 110000. Hence, the entire line of the field can be split into seven intervals—[0, 3], [4, 7], [8, 15], [16, 27], [28, 31], [32, 47], and [48, 63]—versus the eight regular intervals in the HiCuts algorithm. The same processing can be repeated for the field. The starting boundaries in the and fields are noted in Fig. 1. The decision tree of the proposed algorithm that makes cuts according to the starting rule boundaries is shown in Fig. 4. Here, *binth* is also set as three. At the root node, the field is used to split the entire line to seven disjointed intervals; hence, the root node has seven binary search entries. At level 2, the field is used to further split each subspace that has more rules than *binth*. Comparison of the proposed decision tree to the HiCuts decision tree shows that the BC algorithm does not cause unnecessary cutting. Clearly, no two leaves have the same set of rules. Since cutting does not occur in the absence of a boundary, unnecessary cutting is avoided. Hence, rule replication is reduced in the BC algorithm. Additionally, the depth of the decision tree is always less than or equal to six including a leaf since each field is used once at the most.

2) Searching in the BC Algorithm:

The cuts at each inside hub of the BC choice tree don't have settled interims. Thus, at each inner hub of the tree, a paired inquiry is required to decide the best possible edge to take after for given information. Be that as it may, it will be appeared in Section VI that the BC calculation gives preferred pursuit execution over the HiCuts calculation in spite of the memory access for the paired hunt at the inside hubs. Amid the parallel hunt, the pointer to the tyke hub is recalled when the info coordinates the passage esteem or when the information is bigger than the section esteem [20]. Consider an info bundle with headers (000110, 111100, 19, 23, TCP), for instance; subsequent to is utilized at the root hub, a paired pursuit utilizing the header of the given information is performed.

The header 000110 is contrasted with the centre passage of the root hub, which are 010000. Since the info is littler, the pursuit continues to the littler half and looks at the contribution to the section 000100. Since the info is bigger, the kid pointer (the second edge) is recollected, and the pursuit continues to a bigger half. The information is contrasted with 001000, and it is observed to be littler, yet there is no section to continue in a littler half. Subsequently, the hunt takes after the recollected pointer, the second edge. At the second level, by playing out a paired hunt, the last edge is chosen for the header 111100. The linear search, which is the same as that in the HiCuts or Hyper Cuts algorithm, is performed for rules stored in the leaf node. The binary search at each internal node takes time finding a suitable edge to follow for entries.

VI.MODULE DESIGN AND ORGANIZATION

6.1 MODULES:

- Building a BC Decision Tree
- Searching in the Boundary Cutting
- Selective Boundary Cutting
- Data Structure

1) Building A Bc Decision Tree

When the cutting of a prefix plane according to rule boundaries is performed, both the starting and the ending boundaries of each rule can be used for cutting, but cutting by either is sufficient since decision tree algorithms generally search for a subspace in which an input packet belongs and the headers of the given input are compared for entire fields to the rules belonging to the subspace (represented by a leaf node of the decision tree).

2) Searching In the Boundary Cutting

The cuts at each internal node of the bc decision tree do not have fixed intervals. Consequently, at each inward hub of the tree, a parallel inquiry is required to decide the best possible edge to take after for a given info.

Amid the twofold hunt, the pointer to the youngster hub is recollected when the info coordinates the passage esteem or when the information is bigger than the section esteem. Consider an info parcel with headers (000110, 111100, 19, 23, TCP), for instance; subsequent to be utilized at the root hub, a twofold pursuit utilizing the header of the given information is performed. The header 000110 is contrasted with the centre passage of the root hub, which is 010000. Since the information is littler, the hunt continues to the littler half and analyzes the contribution to the passage 000100. Since the input is larger, the child pointer (the second edge) is remembered, and the search proceeds to a larger half. The input is compared to 001000, and it is found to be smaller, but there is no entry to proceed in a smaller half. Hence, the search follows the remembered pointer, the second edge. At the second level, by performing a binary search, the

last edge is selected for the header 111100. The linear search, which is the same as that in the HiCuts or Hyper Cuts algorithm, is performed for rules stored in the leaf node.

3) Selective Boundary Cutting

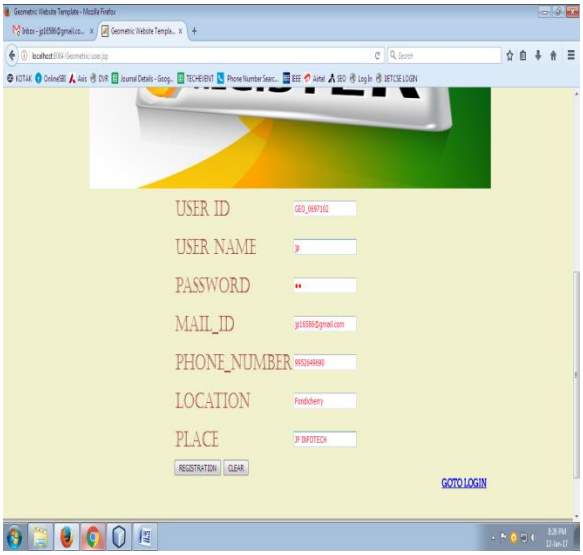
In this module we propose a refined structure for the BC calculation. The choice tree calculations including the BC calculation use binth to figure out if a subspace ought to end up an inward hub or a leaf hub. As such, if the quantity of principles incorporated into a subspace is more than binth, the subspace turns into an interior hub; else, it turns into a leaf hub. In the bc calculation, if a subspace turns into an inner hub, each beginning limit of the guidelines incorporated into the subspace is utilized for cutting. We propose a refined structure utilizing the binth to choose or unselect the limit of a tenet at an inner hub. As it were, the refined structure enacts a tenet limit just when the quantity of guidelines incorporated into an allotment surpasses the binth**.**

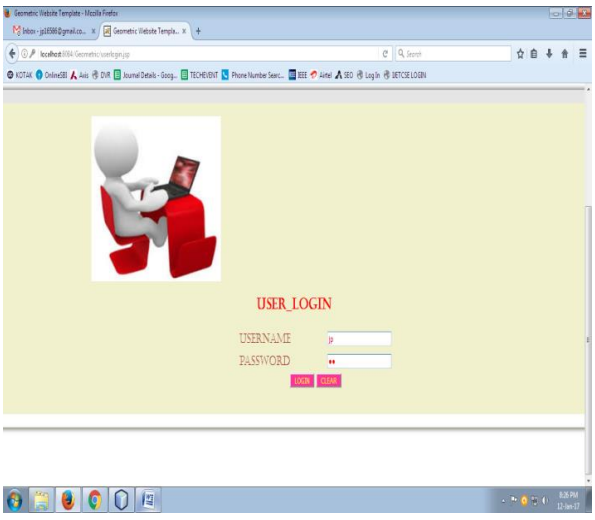
4) Data Structure

There are two distinctive methods for putting away standards in choice tree calculations. The main way isolates a principle table from a choice tree. For this situation, every principle is put away just once in the standard table, while every leaf hub of a choice tree has pointers to the tenet table for the guidelines incorporated into the leaf. The quantity of tenet pointers that every leaf must hold squares with the binth. In looking for the best coordinating guideline for a given bundle or the rundown of every single coordinating principle, after a leaf hub in the choice tree is come to and the quantity of standards incorporated into the leaf is recognized, additional memory gets to are required to get to the tenet table.

The other way includes putting away principles inside leaf hubs. For this situation, look execution is better since additional entrance to the tenet table is stayed away from, yet additional memory overhead is brought about because of standard replication. In our re-enactment in this paper, it is accepted that principles are put away in leaf hubs since the pursuit execution is more critical than the required memory.

VII.SIMULATION RESULTS:





CONCLUSION & FUTURE ENHANCEMENT

We study a general approach to securely search encrypted spatial data with geometric range queries. Specifically, our solution is independent with the shape of a geometric range query. With the additional use of R-trees, our scheme is able to achieve faster-than-linear search complexity regarding to the number of points in a dataset. The security of our scheme is formally defined and analyzed with indistinguishability under Selective Chosen-Plaintext Attacks. Our design has great potential to be used and implemented in wide applications, such as Location-Based Services and spatial databases, where the use of sensitive spatial data with a requirement of strong privacy guarantee is needed.

REFERENCES

[1] B. Chazelle, "Filtering search: A new approach to query-answering," *SIAM J. Comput.*, vol. 15, no. 3, pp. 703–724, 1986.
[2] P. K. Agarwal and J. Erickson, "Geometric range searching and its relatives," *Discrete Comput. Geometry*, vol. 223, pp. 1–56, 1999.
[3] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing," in *Proc. NDSS*, 2011.
[4] H. Shirani-Mehr, F. Banaei-Kashani, and C. Shahabi, "Efficient reachability query evaluation in large spatiotemporal contact datasets," *Proc. VLDB Endowment*, vol. 5, no. 9, pp. 848–859, 2012.
[5] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Berlin, Germany: Springer-Verlag, 2008.
[6] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. Theory Cryptogr. (TCC)*, 2007, pp. 535–554.
[7] E. Shi, J. Bethencourt, T.-H. H. Chan, D. Song, and A. Perrig, "Multidimensional range query over encrypted data," in *Proc. IEEE SP*, May 2007, pp. 350–364.
[8] Y. Lu, "Privacy-preserving logarithmic-time search on encrypted data in cloud," in *Proc. NDSS*, 2012, pp. 1–17.
[9] B. Wang, Y. Hou, M. Li, H. Wang, and H. Li, "Maple: Scalable multidimensional range search over encrypted cloud data with tree-based index," in *Proc. ACM ASIA CCS*, 2014, pp. 111–122.
[10] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proc. ACM SIGMOD*, 2004, pp. 563–574.
[11] R. A. Popa, F. H. Li, and N. Zeldovich, "An ideal-security protocol for order-preserving encoding," in *Proc. IEEE SP*, May 2013, pp. 463–477.
[12] F. Kerschbaum and A. Schropfer, "Optimal average-complexity ideal security order-preserving encryption," in *Proc. ACM CCS*, 2014, pp. 275–286.
[13] B. Wang, Y. Hou, M. Li, H. Wang, H. Li, and F. Li, "Tree-based multidimensional range search on encrypted data with enhanced privacy," in *Proc. SECURECOMM*, 2014, pp. 1–25.
[14] E.-O. Blass, T. Mayberry, and G. Noubir, "Practical forward-secure range and sort queries with update-oblivious linked lists," in *Proc. PETS*, 2015, pp. 81–98.
[15] B. Wang, M. Li, H. Wang, and H. Li, "Circular range search on encrypted spatial data," in *Proc. IEEE ICDCS*, Jun./Jul. 2015, pp. 794–795.
[16] [Online]. Available: http://aws.amazon.com/solutions/case-studies/
[17] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE SP*, May 2000, pp. 44–55.
[18] C. Shahabi, L. Fan, L. Nocera, L. Xiong, and M. Li, "Privacy-preserving inference of social relationships from location data: A vision paper," in *Proc. ACM SIGSPATIAL GIS*, 2015, pp. 1–4.

[19] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*. Boca Raton, FL, USA: CRC Press, 2007.
[20] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. EUROCRYPT*, 2004, pp. 506–522.
[21] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc. ACM CCS*, 2006, pp. 79–88.
[22] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proc. ACM CCS*, 2012, pp. 965–976.