DES SECURED K-NN QUERY OVER SECURE DATA IN CLOUDS

[1]RANJEETH KUMAR M, [2]N SRINIVASU, [3]LOKANATHA C REDDY

[1]Research Scholar, Dept. of CSE, KL University, Guntur, A.P., India
[2]Professor, Dept. of CSE, KL University, Guntur, A.P., India
[3]Professor, Dept. of CS, Dravidian University, Kuppam, A.P., India
E-mail: [1]maduri.ranjith@gmail.com, [2]srinivasu28@kluniversity.in,
[3]lokanathar@yahoo.com

ABSTRACT

Protecting databases or data contents from the web world environment is a tough task for a company. Because every Company/ Financial Institute/ Hospital was hiding their customers or end users list secretly and will not open for all. But now Tom's gang (Hackers) made this possible and tries stealing the data and major portion. In these conditions securing the data outsourcing area such as web hosting and cloud space storage option are becoming very prominent. To manage the situation many were out with secured sharing solutions. Now one more novel approach with high secured and efficient sharing option in data retrieving by end user is demonstrating in this paper. The technique is comprises with two famous algorithms one is DES an encryption scheme and the next is K-NN query passing and data retrieving code.

Keywords: *Tom's Gang, Cloud Spacing, Secured, Sharing, DES, K-NN, Query.*

1. INTRODUCTION

Data outsourcing in cloud is a very challenging task, as conventional encryption does not support processing on top of cipher texts, whereas more recent cryptographic tools such as homomorphism encryption are not flexible enough(they support only restricted operations),and they are also prohibitively expensive for practical uses. To address this problem, previous work such as has proposed privacy-preserving data transformations that hide the data while still allowing the ability to perform some geometric functions evaluation. However, such transformations lack the formal security guarantees of encryption. Other methods employ stronger-security transformations, which are used in conjunction with dataset partitioning techniques, but return a large number of false positives, which is not desirable due to the financial considerations outlined earlier.

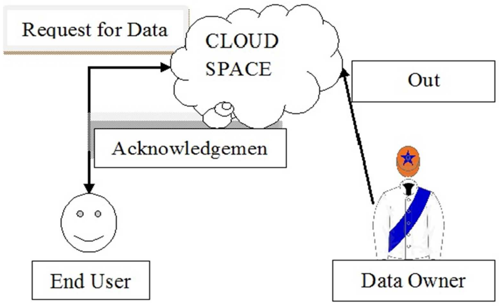

Figure 1: Basic Data Outsourcing In Cloud Space.

2. MATERIALS AND METHODS

Due to the specificity of such data, collecting and maintaining such information is an expensive process, and furthermore, some of the data may be sensitive in nature. For instance, certain activist groups may not want to release their events to the general public, due to con-cerns that big corporations or oppressive governments may intervene and compromise their activities. Similarly, some groups may prefer to keep their geo-tagged datasets confidential, and only accessible to trusted subscribed users, for the fear of backlash from more conservative population groups. It is therefore important to protect the data from the cloud service provider. In addition, due to financial considerations on behalf of the data owner, sub-scribing users will be billed for the service based on a *pay-per-result*model. For instance, a subscriber who asks for *k*NN results will pay for *k*items, and should not receive more than *k*results. Hence, approximate querying methods with low precision, such as existing techniques that return many false positives in addition to the actual results, are not desirable.

Query processing that preserves both the data privacy of the owner and the query privacy of the client is a new research problem. It shows increasing importance as cloud computing drives more businesses to outsource their data and querying services. However, most existing studies, including those on data outsourcing, address the data privacy and query privacy separately and cannot be applied to this problem.

Mobile devices with geo-positioning capabilities (e.g., GPS) enable users to access information that is relevant to their present location. Users are interested in querying about points of interest (POI) in their physical proximity, such as restaurants, cafes, ongoing events, etc. Entities specialized in various areas of interest (e.g., certain niche directions in arts, entertainment, travel) gather large amounts of geo-tagged data that appeal to subscribed users. Such data may be sensitive due to their contents. Furthermore, keeping such information up-to-date and relevant to the users is not an easy task, so the *owners* of such datasets will make the data accessible only to paying customers. Users send their current location as the query parameter, and wish to receive as result the nearest POIs, i.e., nearest-neighbors (NNs). But typical data owners do not have the technical means to support processing queries on a large scale, so they outsource data storage and querying to a *cloud service provider*. Many such cloud providers exist who offer powerful storage and computational infrastructures at low cost. However, cloud providers are not fully trusted, and typically behave in an *honest-but-curious* fashion. Specifically, they follow the protocol to answer queries correctly, but they also collect the locations of the POIs and the subscribers for other purposes. Leakage of POI locations can lead to privacy breaches as well as financial losses to the data owners, for whom the POI dataset is an important source of revenue. Disclosure of user locations leads to privacy violations and may deter subscribers from using the service altogether. In this paper, we propose a family of techniques that allow processing of NN queries in an untrusted outsourced environment, while at the same time protecting *both* the POI and querying users' positions. Our techniques rely on *mutable order preserving encoding* (*mOPE*), the only secure order-preserving encryption method known to-date, also provide performance optimizations to decrease the computational cost inherent to processing on encrypted data, and consider the case of incrementally updating datasets and presenting an extensive performance evaluation of our techniques to illustrate their viability in practice.

In cloud computing, data owner use data and querying services for outsourcing on the cloud data. During this process, data is the separate and private asset of the data owner, hence that must be protected against cloud and querying client. Query which is fired by the client may disclose the sensitive details/information of the client. Hence should be protected in cloud and from data owners. Therefore, one of the major problem in cloud computing is to protect both, data privacy and query privacy amongst the data owner, the client, and the cloud refer Fig- 1. The social networking is one of the rising sectors facing such type of privacy problem [2]. Cloud Computing is new platform to deploying, managing, and providing solution to the various types of storage, platform problems using internet-based processing. However, it is very sensitive issue to upload our personal data on the cloud because data privacy is the big issue and major problem of security. Sensitive information has to be encrypted before outsourcing, which creates the effective infrastructure. The services such as Goggle Docs, Amazon EC2,Microsoft Azure, and Online file storage etc. are the examples of cloud computing and they are widely used by many people worldwide. data utilization services and that is really big challenging task. One of the techniques of retrieval called Symmetric Searchable Encryption (SSE) of encrypted data on the cloud but still there is leakage of data privacy. Secure server –side ranking, which is based on the order-preserving Encryption (OPE), also includes the similarity relevance and robustness [3]. For the privacy of the data, various general solution in recently research papers are deposited to show study on the data privacy, the most general solution in recently done research papers are encryption. It means data deposited service provider must be encrypted to avoid information leakage on the cloud. Agrawal et al [4] proposed one of the solutions so as to order preserving encryption scheme (OPES) by which, indexes can be built directly on cipher text. The various SQL statements such as MAX, MIN, COUNT, GROUP BY and ORDER BY can then be rewritten and processed over the encrypted data. But OPES does not support SUM or AVG statements, in case of SUM and AVG original data must be decrypted first. In private Information retrieval (PIR) for hiding a user's query completely and providing strong privacy and confidentiality, query anonymisation usually uses k-Anonymity [5] and its variants to mix the user's query with other noisy query data. In [6], [7], user privacy and data privacy is considered together. Yonghong Yu and WenyangBai discussed how to enforce data privacy and user privacy over outsourced database service in [8]. Hu et al. [9] proposed one of the solution based on secure traversal framework and privacy homomorphism based encryption scheme.

3. PROPOSING SCHEME

In this paper, we propose a family of techniques that allow processing of NN queries in an untrusted out-sourced environment, while at the same time protecting *both* the POI and querying users positions. Our techniques rely on *mutable order preserving encoding(mOPE)*,which guarantees *indistinguishability under ordered chosen-plaintext attack*(*IND-OCPA*), also provide per-formance optimizations to decrease the computational cost inherent to processing on encrypted data and consider the case of incrementally updating datasets. Inspired by previous work in that brought to-gether encryption and geometric data structures that enable efficient NN query processing and investigate the use of Voronoi diagrams and Delaunay triangulations to solve the problem of secure outsourced *k*NN queries and emphasize that previous work assumed that the contents of the Voronoi diagrams is available to the cloud provider in plaintext, whereas in our case the processing is performed entirely on ciphertexts, which is a far more challenging problem. Our proposed methods for secure nearest-neighbor evaluation perform query processing on top of encrypted data, and for this reason they are inherently expensive. It is a well-known fact that achieving security by processing on encrypted data comes at the expense of significant computational overhead. Next, proposing two optimizations that aim at reducing this cost and secure protocols for processing k-nearest-neighbor queries (kNN) on R-tree index is given. In the authors following work [7], they integrated indexing techniques with secure multiparty computation (SMC) based protocols to construct a secure index traversal framework.In this framework, the service provider cannot trace the index traversal path of a query during evaluation, and hence keep privacy of users. Their protocols for query are complex, and hard to implement. To solve private processing of more specific queries, different techniques have been implemented, e.g. public data column and private data column are implemented by hashing in. But join by hashing is unable to retrieve other specific as well as relevant data columns. Some time before a paper published by researchers proposes kNN queries by processing private & remotely using homomorphism encryption [2]. Theoretical protocols using homomorphic encryption have been proposed to process private document search by specific keywords in a line of documents .These protocols are still too costly to use practically, and they perform only approximated search. Finally, not concerned to private query processing on outsourced encrypted data although our data bucketization is inspired by the data bucketization idea in a work from that area [12].Our approach may also apply to protect query privacy in outsource scenarios.

Spatial database is a database that is optimized to store and query data that represents objects defined in a geometric space. Most spatial databases allow representing simple geometric objects such as points, lines and polygons. Some spatial databases handle more complex structures such as 3D objects, topological coverages, linear networks, and TINs. While typical databases are designed to manage various numeric and character types of data, additional functionality needs to be added for databases to process spatial data types efficiently.

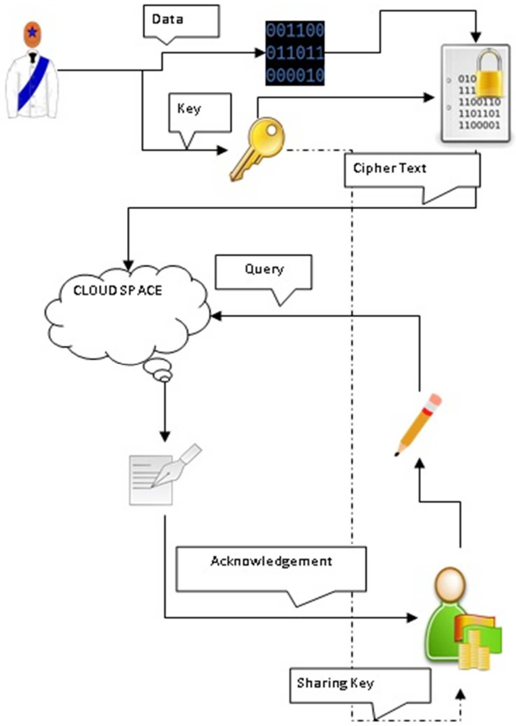

Figure 2: Block Diagram Of Proposing Scheme

As mentioned previously, the dataset of points of interest represents an important asset for the data owner, and an important source of revenue. Therefore, the coordinates of the points should not be known to the server and assume an *honest-but-curious* cloud service provider. In this model, the server executes correctly the given protocol for processing *k*NN queries, but will also try to infer the location of the data points. It is thus necessary to encrypt all information stored and processed at the server. To allow query evaluation, a special type of encryp-tion that allows processing on cipher texts is necessary. In our case, we use the mOPE technique from [6].mOPE is a provably secure order-preserving encryption method, and our techniques inherit the IND-OCPA security guarantee against the honest-but-curious server provided by mOPE. Furthermore, assume that there is no collusion between the clients and server, and the clients will not disclose to the server the encryption keys.

The server receives the dataset of points of interest from the data owner in encrypted format, together with some additional encrypted data structures (e.g., Voronoi diagrams, Delaunay triangulations) needed for query processing.The server receives *k*NN requests from the clients, processes them and returns the results. Although the cloud provider typically possesses powerful computational resources, processing on encrypted data incurs a significant processing overhead, so performance considerations at the cloud server represent an important. The client has a query point *Q*and wishes to find the point's nearest neighbors. The client sends its encrypted location query to the server,and receives *k*nearest neigh-bors as a result.Note that, due to the fact that the data points are encrypted, the client also needs to perform a small part in the query processing itself, by assisting with certain steps.

Focus on securely finding the 1NN of a query point. Employ Voronoi diagrams [1], which are data structures especially designed to support NN queries. An example of Voronoi diagram is shown in Figure 3. Denote the Euclidean distance between two points *p* and *q* by(*p*,*q*), and let *P*={*p* ,*p* ,…,*p* } be a set of *n* distinct points in the plane. Answering a 1NN query boils down to checking which Voronoi cell contains the query point. In our system model, both the data points and the query must be encrypted. Therefore, it needs to check the enclosure of a point within a Voronoi cell securely. Next, propose such a secure enclosure evaluation scheme.

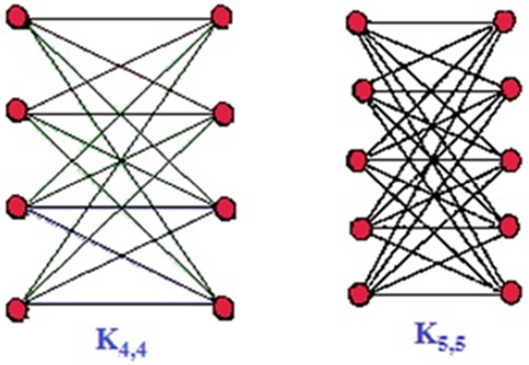

Figure 3: K-Nn Clustering Diagram

Data Owner sends to Server the encoded Voronoi cell vertices coordinates, MBR boundaries for each cell, encoded right-hand side, and encrypted, for each cell edge. Client sends its encoded query point to the Server. Server performs the filter step, determines for each kept cell the edges that intersect the vertical line passing through the query point and sends the encrypted slope, of the two edges to the Client. Client computes the left-hand side, encodes it and sends it to the Server. Server finds the Voronoi cell enclosing the query point and returns result to Client.

To support secure *k*NN queries, where *k* is fixed for all querying users, could extend the VD-1NN method from by generating order-*k* Voronoi diagrams .However, this method, which is called VD-*k* NN, has several serious drawbacks:

(1) The complexity of generating order- *k* Voronoi diagrams is either depending on the approach used or significantly higher than for order-1 Voronoi diagrams.

(2) The number of Voronoi cells in an order- *k*-Voronoi diagram, or roughly when *k*<<*n*. That leads to high data encryption over head at the data owner, as well as prohibitively high query processing time at the server (a *k*-fold increase compared to VD-1NN). Motivated by these limitations of VD-*k*NN, we first intro-duce a secure distance comparison method (SDCM).

Next, devise Basic *k*NN (B*k*NN), a protocol that uses SDCM as building block, and answers *k*NN queries using repetitive comparisons among pairs of data points. B*k*NN is just an auxiliary scheme, very expensive in itself, but it represents the starting point for Triangulation *k*NN (T*k*NN), presented T*k*NN builds on the B*k*NN concept and returns exact results for *k*=1. For *k*>1, it is an approximate method that provides high-precision *k*NN results with significantly lower costs.

Algorithm for data encryption with *k*NN

Input:

Data to be encrypted (Di)

Output:

```
1: Every 8th bit of the unknown key is an odd parity
2: Remove parity based on permutations
    i.e, i= first bit of last byte of 8 bytes
    (Ki),K(i-8),….K(i+1),    K((i+1)-8)….,    K(i+2),
    K((i+2)-8),… key permutated bits after removing
    parity bits
3: Split keys into right block and left block for the
    remaining 36 bits L(0)=P(1:28) R(0)=P(29:56)
4: For 1<=i<=16 (i.e ., 16 sub keys need to generate)
5: Applying left circular shift to generate 16 sub keys
6: L(i)=LS[i] L[i-1]
7: R(i)= LS[i] R[i-1]        //here LS is left shift
8: K[i]=P2[C(i) D(i)]
9: End for
10: Process 64 data bits (db)
```

```
11: Permutation of db result in
    J= second bit of last byte of 8 bytes
    Db= db(j) d(j-8) …. d(j+2) , db((j+2)-8) , ….
12: Split db into L and R blocks
    dl(0)=dp(1:32)
    dr(0)= dp(33:64)[i-1]
13: For i<=1 <=16
14: dl[i]=R[i-1]
15: dr[i]=L[i-1] XOR F(R[i-1],k[i])
16: Chipper text is cb=PP[(drc16) d(c16)]
```

4. RESULTS

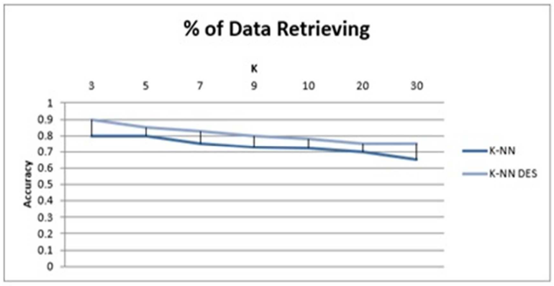

Figure 4: Chart for Accuracy.

As the number clusters increasing the data size get increasing so the retrieving possibility is getting reduced time by time. Even the time of execution is getting increased, so to push these entire flaws aside and trying to attain data retrieving condition formidably. But this needs to be more relevant while passing the query an encrypted will find less in size than a normal. So, the encrypted data is easily retrievable and due cluster framing accurate results are obtained all these were tested under different modes of uploading, sharing, encrypting, decrypting and downloading the data with in different cluster. This helps in obtaining an average accuracy of above 80% for overall performance under different cluster schemes. Experimental results for accuracy versus various clusters as follows.

Table 1: Different Clusters Versus Accuracy.

CLUSTERS	K-NN in %	K-NN DES in %
3	80	90
5	80	85
7	75	83
9	73	80
10	72	78
20	70	75
30	65	75

5. CONCLUSION

Acquiring an efficient encrypting scheme with a secured data retrieving scheme provides a catchment proof data for analyses and this helps in not only providing security to data but also helps in acquiring the data speedily in terms of time and execution in a cloud storage. So, the possibility of data hacking and passing query was very much easy in these arenas.

REFERENCES:

[1] YousefElmehdwi, Bharath K. Samanthula and Wei Jiang, *"Secure k-Nearest Neighbor Query over Encrypted Data in Outsourced Environments",* Technical Report, Department of Computer Science, Missouri S&T, July 2013.

[2] Hu, Haibo, et al. *"Processing private queries over untrusted data cloud through privacy homomorphism"*, Data Engineering (ICDE), 2011 IEEE 27th International Conference on.IEEE, 2011.

[3] Nandhini, N., and P. G. Kathiravan. *"An Efficient Retrieval of Encrypted Data in Cloud Computing."* IJIRCCE, Vol.2, Special Issue 1, March 2014, pages 474-482, Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India.

[4] RakeshAgrawal, Jerry Kiernan, Ramakrishnan Srikant, and YirongXu.Order *preserving encryption for numeric data. In Proceedings* of the 2004 ACM SIGMOD international conference on Management of data, SIGMOD '04, pages 563–574, New York, NY, USA, 2004, ACM.

[5] P. Samarati and L. Sweeney.*Protecting privacy when disclosing information: k-anonymity and*

*its enforcement through generalization and suppression.*Technical report, 1998.

[6] TingjianGe, Stanley B. Zdonik, and Stanley B. Zdonik, *Answering aggregation queries in a secure system model.* In VLDB, pages 519–530, 2007.

[7] HaiboHu and JianliangXu.*Non-exposure location anonymity*. In Yannis E. Ioannidis, DikLun Lee, and Raymond T. Ng, editors, ICDE, pages 1120–1131. IEEE,2009.

[8] Yonghong Yu and WenyangBai, *Enforcing data privacy and user privacy over outsourced database service.* JSW, 6(3):404–412, 2011.

[9] HakanHacgm, BalaIyer, and SharadMehrotra. *Efficient execution of aggregation queries over encrypted relational databases.* In Yoon Joon Lee, Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, Database Systems forAdvanced Applications, volume 2973 of Lecture Notesin Computer Science, pages 125–136, Springer Berlin Heidelberg, 2004.

[10] Varghese, Jiss, and Lisha Varghese. *"Homomorphic Encryption for Multi-keyword based Search and Retrieval over Encrypted Data."* IJAIEM, Volume 3, Issue 8, August 2014, pages 138-146.

[11] C. Gentry, *Fully homomorphic encryption using ideal lattices.* In STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing, pages 169– 178, 2009.

[12] Josep Domingo-Ferrer, *A provably secure additive and multiplicative privacy homomorphism.* In Proc. 5th International Conference on Information Security, 2002 Proc. 5th International Conference on Information Security, 2002

.