Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 7, July 2014, pg.230 – 238

RESEARCH ARTICLE

PRIVACY ON METRIC DATA ASSETS

Dr. M.V. Siva Prasad, G. Sreenivasa Rao, L. Ashok

CSE, Anurag Engineering College, India

CSE, Anurag Engineering College, India

CSE, Anurag Engineering College, India

principal@anurag.ac.in, hod.cse@anurag.ac.in, ashok.it1225@gmail.com

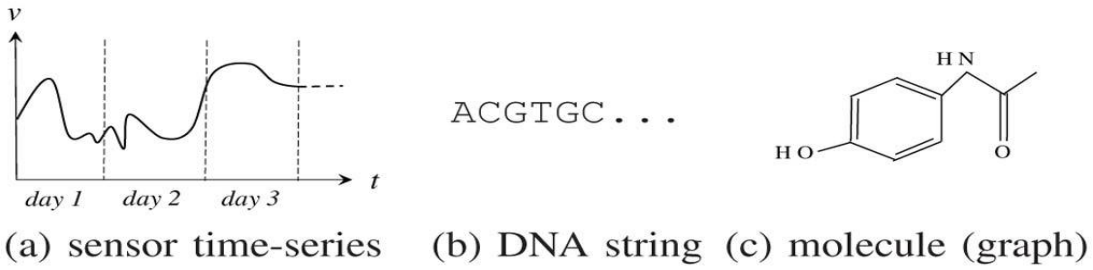
Abstract— ***This paper considers a cloud computing setting in which similarity querying of metric data is outsourced to a service provider. The data is to be revealed only to trusted users, not to the service provider or anyone else. Users query the server for the most similar data objects to a query example. Outsourcing offers the data owner scalability and a low-initial investment. The need for privacy may be due to the data being sensitive (e.g., in medicine), valuable (e.g., in astronomy), or otherwise confidential. Given this setting, the paper presents techniques that transform the data prior to supplying it to the service provider for similarity queries on the transformed data. Our techniques provide interesting trade-offs between query cost and accuracy. Various techniques are built to transform and encrypt data before storing to database server for security reasons. The queries made by trusted clients are also protected in the same fashion. They offer an intuitive privacy guarantee. Empirical studies with real data demonstrate that the techniques are capable of offering privacy while enabling efficient and accurate processing of similarity queries.***

Keywords— Query processing, Security, integrity, and protection

I. INTRODUCTION

ADVANCES in digital measurement and engineering technologies enable the capture of massive amounts of data in fields such as astronomy, medicine, and seismology. The effort of data collection and processing, as well as its potential utility for research or business, creates value for the data owner. He wishes to store them and allow access by himself, colleagues, and other (trusted) scientists or customers. This can be supported by outsourced servers that offer low storage costs for large databases. For instance, outsourcing based on cloud computing is becoming increasingly attractive, as it promises pay-as-you-go, low storage costs as well as easy data access. However, care needs to be taken to safeguard data that are valuable or sensitive against unauthorized access. In this context, we call any item in a data collection an object, individuals with authorized access query users, and the entity offering the storage service the service provider. To analyse the data, authorized scientists may search for similar patterns in collected time series, such as certain daily or hourly sub sequences that indicate interesting phenomena. In this scenario, time series can be represented as vectors of values in chronological order (see Fig. 1a). At query time, a user specifies an example time series q and wishes to obtain those time series most similar to q; the system then retrieves the time series p in the database with the minimum distance to q.

© 2014, IJCSMC All Rights Reserved



(a) sensor time-series (b) DNA string (c) molecule (graph)

FIG. 1. APPLICATION EXAMPLES OF VALUABLE DATA.

Many applications in science and business rely on similarity search of metric data other than time series and vector data. Computer-aided gene sequencing uses the similarity between an unknown sequence from one species and a known sequence from a closely related species to predict the former's function4 (see Fig. 1b). In drug design, pharmacists search for the most similar graph structures to their quest for a suitable molecule, which can be represented as a labeled graph,5 as shown in Fig. 1c.

Objective: The goal of this research is to develop a transformation method tðÞ for converting an original object p in a metric space into another metric space object p0 ¼ tðpÞ. First, the data owner specifies a key value CK in order to define the instance of tðÞ to be used. In a preprocessing phase, the data owner computes p0 for each object p and uploads it to the server (i.e., service provider). At query time, the query user specifies his query object q and then submits the transformed query object q0 to the server for similarity search. The transformation method must satisfy these requirements:

- Even in the worst case that the attacker knows the inverse of tðÞ, he can only estimate the original object p from the transformed object tðpÞ with bounded precision.
- It enables high-query accuracy.
- It enables efficient query processing in terms of communication cost.
- It supports insertion and deletion of objects.

Our contributions are as follows: We present three transformation techniques that satisfy the above requirements. They represent various trade-offs among data privacy and query cost and accuracy.

- In our first solution, we propose an encrypted index-based technique with perfect privacy, but multiple communication rounds. This technique flexibly reduces round trip latency at the expense of data transfer.
- For our second solution, our private anchor-based indexing guarantees the correct answer within only 2 rounds of communication. Retrieval is accelerated by bounding the range of potential nearest neighbors (NN) in the first phase.
- Our third solution limits communication to a single round, and also returns a constant-sized candidate set by computing a close approximation of the query result.
- We extend our solutions in order to meet an intuitive privacy guarantee requirement.

II. RELATED WORK

Privacy and Security: The various techniques have been developed to maintain the confidentiality of outsourced data. Given a relational table] map the tuples of the table into buckets and then store the encrypted tuples of those buckets at the server. At query time, the user compares the query object against the description of those buckets, and then determines the necessary buckets that need to be retrieved from the server. In another proposal, the data owner applies the encryption function on each node separately and then stores all encrypted tuples at the server. The method employs an order-preserving function on 1D data values such that the distribution of output values is different from that of input values. The two works present several transformation-based techniques for outsourcing spatial data to the (untrusted) server, such that the server is able to perform spatial range search correctly for trusted users on those transformed points, without knowing their actual coordinates. They propose spatial transformations in 2D space based on scaling, shifting, and noise injection. Also, they develop a solution using an encrypted R-tree. Those solutions operate on explicit 2D coordinates, rendering them inapplicable in our setting, where the distance function is a generic distance metric. propose to outsource multidimensional points to the (untrusted) server, by using a secure scalar product encryption technique. Methods are then provided for kNN search at the server, without the server learning the distances among the points. However, the secure scalar product relies on specific properties of the Euclidean distance in the multidimensional space. It is not applicable to other Lp norms, e.g., the L1 norm (the Manhattan distance). Obviously, it also cannot be applied to our problem setting which considers arbitrary metric space objects (e.g., strings, graphs, time-series). Another drawback of this proposal is that no indexing scheme can be built on the encrypted tuples, forcing the server to perform a linear scan over the data set. This affects severely

© 2014, IJCSMC All Rights Reserved

the scalability of the system. In the field of privacy-preserving data mining, perturbation techniques have been developed for introducing noise into the data, before sending them to the service provider. However, such an approach does not guarantee the exact retrieval of results. The k-anonymity model has been applied extensively for the privacy-preserving publication of data sets. The idea is to generalize the tuples in a table such that each generalized representation is shared by at least k tuples. This way, each object cannot be distinguished from at least k _ 1 other objects. It is often used to generalize the medical records of patients so that the adversary cannot link a specific patient to a medical record. Except for some person-related data like DNA data, most of the metric data that we consider (e.g., astronomy data, time series) is collected from nature rather than from persons.

A.PROBLEM SETTING

We start by introducing our scenario and our problem setting in Section 3.1. Then, we propose an intuitive privacy guarantee for metric data in Section 3.2. Next, we describe straightforward solutions to our problem and discuss their drawbacks in Section 3.3.

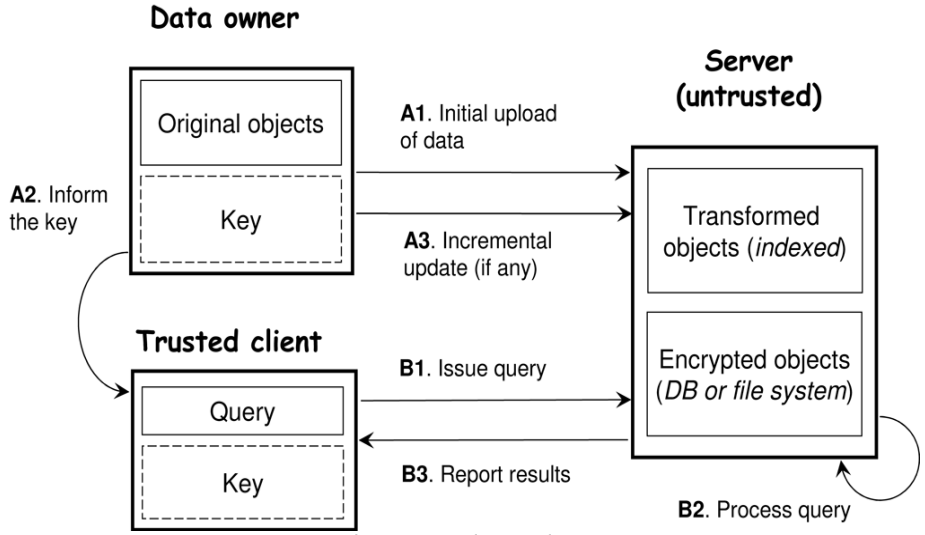


Fig. 2. Scenario overview.

B. PROBLEM DEFINITION

We first discuss our scenario and then define our problem.

a). Scenario: Fig. 2 depicts our scenario for outsourcing data. It consists of three entities: a data owner, a trusted query user, and an untrusted server. On the one hand, the data owner wishes to upload his data to the server so that users are able to execute queries on those data. On the other hand, the data owner trusts only the users, and nobody else (including the server). The data owner has a set P of (original) objects (e.g., actual time series, graphs, strings), and a key to be used for transformation. First, the data owner applies a transformation function (with a key) to convert P into a set P0 of transformed objects, and uploads the set P0 to the server (see step A1 in the figure). The server builds an index structure on the set P0 in order to facilitate efficient search. In addition, the data owner applies a standard encryption method (e.g., AES) on the set of original objects; the resulting encrypted objects (with their IDs) are uploaded to the server and stored in a relational table (or in the file system). Next, the data owner informs every user of the transformation key (see step A2). In the future, the data owner is allowed to perform incremental insertion/deletion of objects (see step A3). At query time, a trusted user applies the transformation function (with a key) to the query and then sends the transformed query to the server (see step B1). Then, the server processes the query (see step B2), and reports the results back to the user (see step B3). Eventually, the user decodes the retrieved results back into the actual results. Observe that these results contain only the IDs of the actual objects. The user may optionally request the server to return the actual objects that correspond to the above result set.

b)Problem Definition:

We will use the term object for the metric data of interest to the data owner. A transformed object then refers to an object obtained from a transformation.

© 2014, IJCSMC All Rights Reserved

TABLE 1
List of Notations

Notation	Meaning
P	the set of original objects
P'	the set of transformed objects
$p.id$	the ID of the object p
$dist(p_i, p_j)$	the distance between objects p_i and p_j
$\mathcal{CK}$	encryption key
$\mathcal{ECR}(X, \mathcal{CK})$	encrypt the object X using the key $\mathcal{CK}$
$\mathcal{DCR}(X, \mathcal{CK})$	decrypt the object X using the key $\mathcal{CK}$
$\mathcal{OPE}(v)$	order-preserving encryption on a data value v
$hamming(\mathcal{B}, \mathcal{B}')$	Hamming distance between bitmaps $\mathcal{B}$ and $\mathcal{B}'$
δ	the value used in the δ-gap guarantee

Let dist(p_i,p_j) denote the distance between two objects p_i and p_j. We focus on nearest neighbor queries, for simplicity. The extension to the case of k nearest neighbors is straightforward. We first give the definition of the nearest neighbor query as follows:

Definition 1 (Nearest Neighbor Query): Given a query object q and a set P of objects, the nearest neighbor query retrieves the object P_{nn}€P such that dist(q,p_{nn})≤dist(q,p) for all p€P.

Recall from Fig. 2 that both steps A1 and B1 require the data owner and the user to apply a transformation function.

Our research objective is to design a transformation method that meets the following requirements:

- Even in the worst case where the attacker knows the inverse of the transformation function, the attacker can only estimate the original object p from the transformed object p_0 with bounded precision.
- It enables high-query accuracy.
- It enables efficient query processing in terms of communication cost.
- It supports insertion and deletion of objects.
-

C. Privacy Guarantee:

In this section, we employ an intuitive obfuscation-based privacy guarantee that can be adapted for metric data. In the two-dimensional space, obfuscation has been used to represent an object's location by a superset region called the obfuscated region. An adversary without a priori knowledge is unable to distinguish the object's actual location from other locations in the obfuscated region. The privacy value is typically expressed as the area of the obfuscated region in the two-dimensional space. However, for generic metric spaces, there is only the concept of distance but not area. Privacy thus means avoiding small distance between an object and its obfuscated representation. We propose to obfuscate an object p by using a ring(a,dist(a,p))whose center is a reference object a and radius is dist(a,p).This way, the object cannot be distinguished from any other possible object (not necessarily from the data set) that have the same obfuscated ring representation. We formally define this privacy guarantee as:

Definition 2 :(δ-Gap Guarantee). Let p be an object of the data set P. The ring (a,dist(a,p))satisfies the δ-gap guarantee if dist(a,p)≥δ

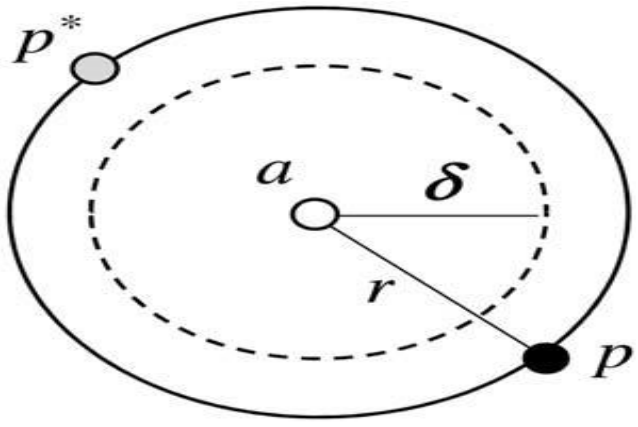


Fig. 3. Distance guarantee: δ-gap.

© 2014, IJCSMC All Rights Reserved

The data owner is able to tune the value of _ such that it describes exactly the required degree of obfuscation. In the example of Fig. 3a, the object p is represented by the ring (a,r) where a is a reference object and r=dist(a,p).Since r≥δ .the ring satisfies the δ-gap guarantee. Observe that any possible object (e.g., object p_) having distance r from object a also has the same representation as (a,r).The reference object a could be either an object picked from the data set P or a randomly generated object that satisfies dist(a,p)=δ.Regarding the choice of δ, we suggest to set δ to the average cth nearest neighbor distance, where c is a small constant. This value causes the objects to be displaced in the vicinity of their neighbors, without significantly affecting most of the distances between pairs of objects. For certain data related to individuals (e.g., DNA data), a more appropriate privacy guarantee may be k-anonymity. Its adaptation to this paper's problem setting and the extensions to the paper's proposals necessary to support it are available in a technical report .

III. PROPOSED APPROACH

A.EHI Searching Algorithm for Client:

Algorithm EHI-Search

1: request the server for the (encrypted) root node L root;

2: H:=new min-heap; p_{nn}:=NULL;
3: γ:=$\min_{e \in L_{root}} maxdist(q,e)$; ▷ derive NN distance bound
4: **for each** entry $e \in L_{root}$ such that $mindist(q,e) \leq \gamma$ **do**
5: insert the entry $\langle e, mindist(q,e) \rangle$ into H;
6: **while** H is not empty and its top entry's key $\leq \gamma$ **do**
7: pop next λ entries from H and insert them into a set S;
8: request the server for each (encrypted) child node of S;
9: **for each** retrieved node L_{cur} **do**
10: **if** L_{cur} is a leaf node **then** ▷ check for closer objects
11: update γ and p_{nn} by using objects in L_{cur};
12: **else** ▷ expand the entries of L_{cur}
13: γ:=$\min\{\gamma, \min_{e \in L_{cur}} maxdist(q,e)\}$;
14: **for each** $e \in L_{cur}$ such that $mindist(q,e) \leq \gamma$ **do**
15: insert the entry $\langle e, mindist(q,e) \rangle$ into H;
16: **return** p_{nn} as the result;

Traditional encryption methods are capable of protecting the confidentiality of the data. However, this also prevents users from querying the data on the untrusted server. Obviously, transferring all the encrypted data to the query user for searching takes outsourcing ad absurdum. Moreover, when services are made available to users on a pay-as you-go basis, the service providers are not interested in such a brute force data transfer. Note that certain applications may involve large numbers of query users from scientific institutions, hospitals, or branch offices globally. For example, consider a query load of 10,000 queries/s. If a brute force solution transfers 10 MBytes for a single query, the server needs a network bandwidth of 100,000 MBytes/s, which is far beyond the available bandwidth of Fast Ethernet (100 Mbits/s). Also, from the user point of view, it is better to have most of the processing done in the cloud. For example, a user with a smart phone may not have enough resources (bandwidth, CPU, battery power) to download and query a large data set locally. Therefore, the design of communication efficient solutions is of critical importance to the success of cloud computing applications, allowing these to be operated at low cost. Typically, cloud computing providers (e.g., Amazon, HP, and Microsoft) attempt to solve the problem by offering contractual agreement that promise not to release outsourced data to third parties. Nevertheless, even if the provider respects the contractual agreement, the data is not guaranteed to be safe. Unintended leaks of data is reported regularly, and hackers may still exploit vulnerabilities to gain access to data. Therefore, we believe that data owners will find it attractive to outsource encrypted rather than plain data. Closest to our work are the recent outsourcing proposals on searching problems in the spatial domain and multidimensional space, respectively. Unfortunately, their techniques rely on specific properties of those spaces and they cannot be extended to solve our problem, which considers arbitrary metric

© 2014, IJCSMC All Rights Reserved

data spaces (e.g., strings, graphs, time-series). The goal of this research is to develop a transformation method tðÞ for converting an original object p in a metric space into another metric space object. First, the data owner specifies a key value CK in order to define the instance of tðÞ to be used. In a preprocessing phase, the data owner computes for each object p and uploads it to the server (i.e., service provider). At query time, the query user specifies his query object q and then submits the transformed query object q to the server for similarity search. The transformation method must satisfy these requirements: Even in the worst case that the attacker knows the inverse of, he can only estimate the original object p from the transformed object with bounded precision. It enables high-query accuracy. It enables efficient query processing in terms of communication cost. . It supports insertion and deletion of objects. Our contributions are as follows: We present three transformation techniques that satisfy the above requirements. They represent various trade-offs among data privacy and query cost and accuracy.

B.Metric Preserving Transformation (MPT):

The basic idea behind MPT is to pick a small subset of the data set P as the set of anchor objects and then assign each object of P to its nearest anchor. For each object p, we compute its distance distðai; pÞ from its anchor ai and then apply an order-preserving encryption function OPE on the distance value. These order-preserving encrypted distances will be stored in the server and utilized for processing NN queries.

Algorithm MPT-Build (Data Set P, Encryption Key CK, Integer A)

1: use a heuristic of Ref. [11] to select a set of A anchor objects from P;
2: Integer B:=$\lceil |P|/A \rceil$;
3: use a heuristic of Ref. [11] to assign each data object of P to an anchor object, subject to the capacity constraint B;
4: for i:=1 to A do
5: let a_i be the i-th anchor object;
6: let $a_i.S$ be the set of objects assigned to the anchor a_i;
7: r_i:=$\max_{p \in a_i.S} dist(a_i, p)$; ▷compute covering radius
8: for each object $p \in a_i.S$ do
9: send the tuple $\langle p.id, \mathcal{OPE}(dist(a_i, p)), \mathcal{ECR}(p, CK) \rangle$ to the server;

\MODULES:

1. Outsourcing Data
2. Nearest Neighbor Query
3. Brute-force Secure Solution (BRUTE)
4. Anonymization - based Solution (ANONY)

.Outsourcing Data:

It consists of three entities: a data owner, a trusted query user, and an untrusted server. The data owner wishes to upload his data to the server so that users are able to execute queries on those data. The data owner trusts only the users, and nobody else (including the server). The data owner has a set P of (original) objects (e.g., actual time series, graphs, strings), and a key to be used for transformation. First, the data owner applies a transformation function (with a key) to convert P into a set P0 of transformed objects, and uploads the set P0 to the server.

Nearest Neighbor Query:

In this module, our research objective is to design a transformation method that meets the following requirements:

1) Even in the worst case where the attacker knows the inverse of the transformation function, the attacker can only estimate the original object p from the transformed object p' with bounded precision.

© 2014, IJCSMC All Rights Reserved

2) It enables high query accuracy.
3) It enables efficient query processing in terms of communication cost.
4) It supports the insertion and deletion of objects.

Brute-force Secure Solution (BRUTE):

In the offline construction phase, the data owner applies conventional encryption (e.g., AES) on each object and then uploads those encrypted objects to the server. At query time, the user needs to download all encrypted objects from the server, decrypt them and then compute the actual result. it is perfectly secure, but its query and communication cost are both prohibitively high, and pay-as you- go is not supported. Anonymization - based Solution (ANONY) This anonymization-based solution achieves data privacy by means of k-anonymity — the objects are generalized in such a way that each generalized object cannot be distinguished from k-1 other generalized objects. In the context of similarity search, it is able to confuse the ranking of transformed objects because k-1 of them have the same rank as the transformed object of the actual nearest neighbor. Thus, we still consider this solution as a competitor, even though k-anonymity is not a suitable privacy guarantee for our applications. 1.1 Shortcomings of Existing Method s In the literature, a number of concepts for securing databases have been studied. Private information retrieval techniques hide the user's query, e.g., the
data item searched for, but not the data being queried. To outsource valuable data to an insecure server, such techniques are clearly not appropriate. Digital watermarking establishes the data owner's identity on the data. Additional information stored in the data helps prove ownership, but it cannot prevent an attacker from illegally copying the data set. Anonymization techniques secure data by releasing only a generalized version. Aggregate statistical analysis is still possible on the generalized data, but the result of a specific query is not guaranteed to be accurate.

IV. EXPERIMENTAL RESULTS

The straightforward approaches BRUTE and ANONY have very high-communication cost when compared to the others. EHI has a moderate communication cost because the client performs the actual search procedure. Observe that MPT and FDH greatly reduce the communication cost by outsourcing search functionality to the server. MPT obtains its reduction in the communication cost by providing the server with information on the relative distances of objects (i.e., order preserving encryption). The best performance in terms of communication cost is achieved by the FDH method. Compact bitmap representatives allow efficient search with very low communication cost. As shown in Table 6, both the construction time and server CPU time (per query) of all our proposed solutions are reasonable.

A. Comparison between DBH and FDH:

The percentage of empty results, average NN distance, and average communication cost on the GFC data set when varying the parameter A. It is worth noticing that DBH may often return empty results when A is above 100 (see the highlighted cells). This is because each hash bucket contains very few objects in those cases. On the other hand, FDH does not return empty result in any case, thanks to its search strategy based on Hamming distances among the bitmaps. The NN distances returned by DBH and FDH are similar, but the communication cost of FDH is much lower than that of DBH. The rank of the result NN on the data sets SHUTL and GFC. For ease of visualization, the ranks are plotted in descending order. Observe that the result rank of FDH outperforms that of DBH on the data sets YEAST, MUSH, and SHUTL. For the data set GFC, the 10 percent of the query instances with that largest ranks favor FDH much more than DBH, but the next 30 percent of the query instances favor DBH slightly more than FDH.

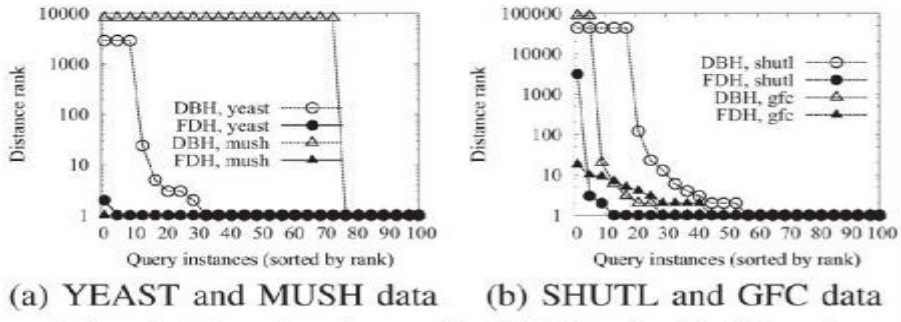


Fig. 4. Rank of result NN for individual queries on real data sets.

B. Effect of the Selection Parameter θ:

The parameter θ in the MPT and FDH techniques to determine a good value that optimizes communication cost. The figure depicts the results: increasing θ means that the client requests more tuples/bitmaps from the server.

© 2014, IJCSMC All Rights Reserved

In MPT, θ is used in the initial communication round. Hence, it influences only the quality of the initial approximation. The better this initial approximation, the lower the communication cost is in subsequent rounds. This is reflected in the experimental results where increasing θ lowers the communication cost of MPT.

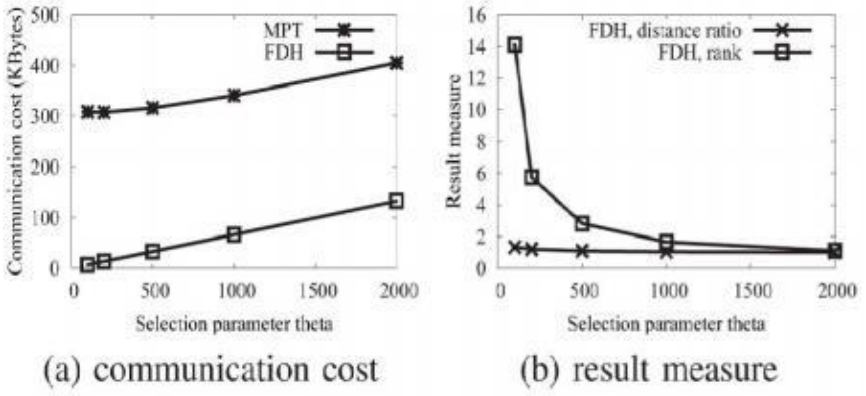


Fig. 5. Effect of θ, on GFC data.

V. CONCLUSION

Proposing similarity search techniques for sensitive metric data bioinformatics data that enable outsourcing of such search. Existing solutions either offer query efficiency at no privacy, or they offer complete data privacy while sacrificing query efficiency. We introduce approaches that shift search functionality to the server. The proposed Metric Preserving Transformation stores relative distance information at the server with respect to a private set of anchor objects. This method guarantees correctness of the final search result, but at the cost of two rounds of communication. The proposed Flexible Distance-based Hashing methods finishes in just a single round of communication, but does not guarantee retrieval of the exact result.

REFERENCES

[1] G. Aggarwal, T. Feder, K. Kenthapadi, S.Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering," Proc. 25th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS), pp. 153-162, 2006.
[2].R.Agrawal and R. Srikant, "Privacy-Preserving Data Mining,"Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 439-450, 2000.
[3]. R.Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order-Preserving Encryption for Numeric Data," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 563-574, 2004.
[4]. N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*- Tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 322-331, 1990.
[5]. S.Berchtold, D.A. Keim, and H.-P. Kriegel, "The X-Tree : An Index Structure for High-Dimensional Data," Proc. 22nd Int'l Conf. Very Large Databases, pp. 28-39, 1996.
[6]. T. Bozkaya and Z.M. O ¨ zsoyoglu, "Indexing Large Metric Spaces for Similarity Search Queries," ACM Trans. Database Systems, vol. 24, no. 3, pp. 361-404, 1999.
[7]. G.R. Hjaltason and H. Samet, "Index-Driven Similarity Search in Metric Spaces," ACM Trans. Database Systems, vol. 28, no. 4, pp. 517-580, 2003.
[8]. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the Privacy Preserving Properties of Random Data Perturbation Techniques," Proc. IEEE Third Int'l Conf. Data Mining (ICDM), pp. 99-106, 2003.
[9]. L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5,pp. 557-570, 2002.
[10]. W.K. Wong, D.W. Cheung, B. Kao, and N. Mamoulis, "Secure k-NN Computation on Encrypted Databases," Proc. 35th ACM SIGMOD Int'l Conf. Management of Data, pp. 139-152, 2009.

AUTHOR PROFILE:

[1].Dr. M.V. Siva Prasad awarded PhD from Nagarjuna University, Guntur, received M.Tech. [SE] from VTU, Belgaum and B.E. [CSE] from Gulbarga University, presently working as principal in Anurag Engineering College (AEC), Ananthagiri(V), Kodad(M), Nalgonda(Dt.), Andhra Pradesh, India

© 2014, IJCSMC All Rights Reserved

[2].G. Sreenivasa Rao received Master of Technology (Computer Science & Engineering) from Jawaharlal Nehru Technological University (JNTUH). My research interests include Information Security, Web Services, Cloud Computing, Data Mining and Mobile Computing. Presently working as Head of The Department for CSE Department in Anurag Engineering College (AEC), Ananthagiri (V), Kodad (M), Nalgonda (Dt.), Andhra Pradesh, India.

[3]. L.Ashok Pursuing Master of Technology (Computer Science & Engineering)from Jawaharlal Nehru Technological University (JNTUH). My research interests include Information Security, Web Services, Cloud Computing, Data Mining and Mobile Computing

© 2014, IJCSMC All Rights Reserved