International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 08, August-2015

Computation of Geographical Distances over Encrypted Coordinates in Cloud Severs

Rana Jassim
Department of Computer Sciences,
Education College for Pure Sciences, Basrah University
Basra , Iraq.

***Abstract* - This paper addresses the problem of how to compute the distance among multiple points on the Earth without compromising the privacy of their geographical locations. A data owner Alice has a set of geographical locations such as military bases. Bob wants to evaluate the distance between his query and the entire set of Alice. The latter outsources the storage of his data to remote cloud servers to enjoy with huge data management services in an efficient cost. For security purposes, he has to encrypt his data prior outsourcing it to the untrusted cloud servers. But, encryption makes computing the encrypted data a challenging task. Our proposed method employs the order preserving symmetric encryption (OPSE) to encrypt the coordinates of geographical points. Such a scheme enables the cloud server to evaluate the geographical distances over encrypted coordinates without decryption.**

Keywords - Cloud computing; Great circle distance; Privacy; OPSE.

I. INTRODUCTION

Due to the great advances in virtualization [1-3] and networking technologies, cloud computing has become more prevalent. Recently, many popular companies like Google, Amazon, and Salesforce have adopted the cloud computing technology to support their customers with a mass storage, large scale data management services in an efficient cost [4]. However, despite its technical advantages and economic benefits, cloud computing introduces new security challenges towards the privacy of users sensitive data [5]. To combat unsolicited access, users usually encrypt their sensitive data before outsourcing it to the cloud servers. However, traditional encryption schemes pose a significant barrier towards computing the encrypted data [6].

Calculating the distance between point locations has many real-world applications such as finding the shortest route among selected airports, or the distance between patients and the hospital. Unfortunately, such a process supposes that geographical location of points is public and thus do not care to the privacy issue. However, in some cases it is desirable to protect locations during the computation process. Consider the following example to see the importance of security issue. Suppose military bases looking for returning the shortest route covering them. For security purposes, no base wants to reveal its location. During the process of identifying the shortest route, it is the best choice for the involved bases not to disclose their locations.

Under the geographic coordinate system [7], the location of each point on the map is specified by two numbers: latitude (φ) and longitude (λ). Both values are an angular measurement, usually expressed in degrees. In this paper, we have used the great-circle distance to measure the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere (as opposed to a straight line through the sphere's interior).

A trivial solution to overcome the security problem is to utilize a trusted third party (TTP). Each party sends his points to TTP. Then the latter investigates their geographical distance. However, in real life situations, finding a completely trusted third party is a difficult task. Our work does not require such a third party.

In this paper, we show how the cloud server evaluates the geographical distance in a privacy preserving way. Given a point *p*, Bob would like to find whether there are points in Alices' collection *D* that are nearest to *p* without disclosing either *p* or *D*. Our notion of geographical distance is based on great circle distance.

The remainder of this paper is organized as follows. Section II discusses the related work. The problem definition is explained in Section III. Section IV introduces the OPSE scheme. We provide our proposed scheme in Section V. Section VI gives performance investigations. Finally, the whole paper is concluded in Section VII.

II. RELATED WORKS

The travelling salesman problem of visiting all 3,810 points in a circuit board was solved using *Concorde TSP Solver* [8]. The computation took approximately 15.7 CPU-years [9]. In May 2004, the traveling salesman problem of visiting all 24,978 points in Sweden was solved. At the time of the computation, this was the largest solved TSP instance, surpassing the previous record of 15,112 points through Germany set in April 2001. The largest solved instance of the traveling salesman problem consists of a tour through 85,900 points in a VLSI application. The computation was carried out with *Concorde TSP Solver* [10].

However, the majority of the above mentioned methods are limited to handling distance in two dimensions. For small areas like cities or counties, this is a reasonable simplification. For longer distances such as those that span larger countries or continents, measures based on two dimensions are no longer appropriate, since they fail to account for the curvature of the Earth. Our work, on the other hand, measures the distance between two points along the surface of the Earth without compromising the privacy of their coordinates.

III. Problem Definition

Our proposed method consists of three parties: Alice (Data owner), Bob (User), and Cloud server. Alice has a set $D = \{v_1, v_2 \ldots, v_n\}$ of n points. Each point v_i is given by its geographical coordinates: latitude (φ_i) and longitude (λ_i). Alice uploads the collection D to the cloud server to enjoy with its technical advantages and economic benefits. During the query time, Bob sends his point b to the cloud server. The latter measures the distance between b and each point in D collection. The great-circle distance is used to measure the proximity between two points.

Revealing the geographical coordinates of some secret points, for example military bases, to the cloud server may jeopardize their privacy. Thus, Alice should encrypt his points before revealing them to the cloud. However, none of the existing encryption methods allow the server to evaluate the distance function over encrypted data. To solve such a problem, we employ the recently developed primitive OPSE to perform such a task.

IV. Order Preserving Symmetric Encryption

OPSE was first developed in the database community for enabling efficient range queries over encrypted data. OPSE is a deterministic encryption scheme that preserves the numerical ordering of its original numbers. Boldyreva and its colleagues [11] designed an efficient scheme. Its security depends on the pseudorandom function's notion. The construction of the proposed scheme is based on the uncovered relation between the random order-preserving function and the *hypergeometric probability distribution* (HGD). The order-preserving function f from domain $D_m = \{1, \ldots, M\}$ to range $R = \{1, \cdots, N\}$, where $N > M$, can be uniquely defined by a combination of M out of N ordered items and fulfill the concept "*as random as possible*". That requires from the attacker to try all the combination of M out of N to break the encryption. The size of the rage R will be discussed later. Algorithm 1 (adopted from [12]) represents the encryption function of OPSE scheme.

Algorithm 1 $\mathrm{OPSE}_{k_1}(D_m, R, m)$

Input: the encryption key k_1, domain D_m, range R, and the plaintext m.
Output: Ct the encrypted number.

$M \leftarrow |D_m|; N \leftarrow |R|$
$d \leftarrow min(D_m) - 1; r \leftarrow min(R) - 1$
$y \leftarrow r + \lceil N/2 \rceil$
if $|D_m| = 1$ **then**
  $coin \leftarrow TapeGen(k_1, (D_m, R, 1||m))$
  $Ct \xleftarrow{coin} R$
  return Ct
end if
$coin \xleftarrow{R} TapeGen(k_1, (D_m, R, 0||m))$
$x \xleftarrow{R} d + Hygeninv(coin, N, y - r, M)$
if $m \leq x$ **then**
  $D_m = \{d + 1, \ldots, x\}$
  $R = \{r + 1, \ldots, y\}$
else
  $D_m = \{x + 1, \ldots, d + M\}$
  $R = \{y + 1, \ldots, r + N\}$
end if
return $\mathrm{OPSE}_{k_1}(D_m, R, m)$
{Where $TapeGen(.)$ is a random number generator. For the sampling function of the HG distribution, we used the $Hygeninv(.)$ efficient MATLAB function.}

V. Proposed Scheme

The work of our proposed scheme is divided into two phases: *indexing and querying*. In the indexing phase, Alice and Bob perform two steps: *coordinates mapping and coordinates encryption*. In the first step, each point is transformed from spherical coordinates into Cartesian ones. In the second step, the coordinates of each point are encrypted before revealing them. In the query phase, the cloud server measures the great circle distance over the encrypted points and returning the distance values back to Bob.

A. Coordinates Mapping

Assume that the Earth is a sphere of radius r. Alice converts the spherical coordinates (φ_i, λ_i) from degree into radian units by multiplying each coordinate with the constant $pi/180$. Then he transforms the points into three-dimensional space as follows [7]:

$$x_i = r\cos\varphi_i \cos\lambda_i$$
$$y_i = r\cos\varphi_i \sin\lambda_i$$
$$z_i = r\sin\varphi_i$$

The great-circle distance d between two points on Earth is defined as a line through three-dimensional space between the points of interest. See Fig.1.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

Fig. 1. Great circle distance

Fig.2a illustrates how to transfer the spherical coordinates of 13 randomly selected points into their corresponding Cartesian ones.

B. *Coordinates Encryption*

This step encrypts the Cartesian coordinates *(x, y, z)* of each point. As explained before OPSE preserves the numerical ordering of its original numbers. Such an amazing properly enables us to conduct the distance function over the encrypted numbers without decryption. Initially, Alice and Bob agree on the initial parameters of the OPSE, which are: secret key k_1, domain D_m, and range R. Then each one encrypts his local coordinates (x_i,y_i,z_i) by using the primitive OPSE, as follows:

$sx_i=OPSE_{k1}(D_m,R,x_i)$

$sy_i=OPSE_{k1}(D_m,R,y_i)$

$sz_i=OPSE_{k1}(D_m,R,z_i)$

The outcome of this step is the encrypted coordinates (sx_i, sy_i, sz_i), $i=1,\ldots, n$. Fig. 2b shows the encrypted coordinates.

(a) Coordinates Mapping

Spherical coordinates		Cartesian coordinates		
φ	λ	x	y	z
119.47	15.02	-4.7516	-1.275	8.7061
82.68	11.56	1.2483	0.25532	9.9185
123.42	46.2	-3.8121	-3.9753	8.3466
76.25	40.17	1.8162	1.5332	9.7134
60.05	38.91	3.8848	3.1357	8.6646
74.34	39.49	2.0831	1.7166	9.6288
13 points 55.84	16.5	5.3838	1.5948	8.2747
88.55	14.02	0.24551	0.061303	9.9968
104.79	22.15	-2.3644	-0.96248	9.6687
97.67	36.52	-1.0726	-0.79427	9.9105
122.11	30.67	-4.5719	-2.7114	8.4703
97.34	22.69	-1.1787	-0.49282	9.9181
122.15	16.45	-5.1036	-1.5069	8.4666

(b) Coordinates Encryption

sx	sy	sz
57.847	80.361	213.7
117.32	115.06	222.61
61.077	61.077	213.7
117.32	117.32	222.61
136.35	136.35	213.7
119.46	117.32	222.61
186.69	117.32	213.7
115.06	115.06	222.61
68.428	89.849	222.61
80.361	89.849	222.61
57.847	68.428	213.7
80.361	89.849	222.61
49.181	80.361	213.7

Fig. 2. Basic steps of the proposed scheme

VI. PERFORMANCE INVESTIGATIONS

In this section, we report the experimental results of our proposed scheme. Our experiments were conducted on a 2.5GHz Intel i5-3210m processor, Windows 7 operating system of 64-bits, with a RAM of 4GB. We used MATLAB R2008a to implement our experiments. The radius of Earth is set to r=6371*km* [13]. The domain of OPSE D_m= [*-r*10 ..... r*10*] and the range R=[*-r*100 ... r*100*]. Table 1 shows the common symbols used in our experiments.

Table I. Symbols used in our experiments

Symbol	Meaning
n	number of points in Alice collection
R	radius of the Earth
x,y,z	Cartesian coordinates
sx, sy, sz	encrypted coordinates
D_m	domain of OPSE
R	range of OPSE

A. *Correctness*

In this experiment, we test the correctness of OPSE to preserve the numerical values of its plaintext numbers. Such an amazing property enables the great circle distance to be performed over the encrypted numbers and returning the correct distances without decryption. Fig. 3 illustrates how the OPSE succeed to preserve the numerical values of the original coordinates.

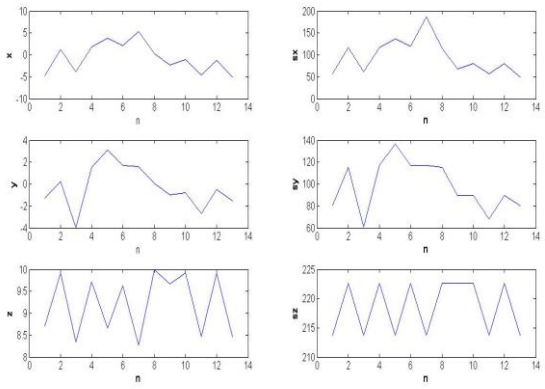

Fig. 3. correctness of OPSE

B. *Encryption Efficiency*

At the point's side, there are two processes: coordinates mapping and coordinates encryption. The latter is the dominant one. As explained in Algorithm 1, OPES works by reducing repeatedly the whole range R to a specific small window and uses the plaintext m as a seed for selecting the cipher text Ct from the last remained window. Recall that our work sets R into the interval [*-r*10, ..., r*10*].

Thus the radius *r* has tight effect on the range *R*. Fig. 4 reports the encryption time as radius increased. As expected, increasing *r* leads to increase the encryption time. This is because a larger range requires more iteration to get the last window in OPSE.

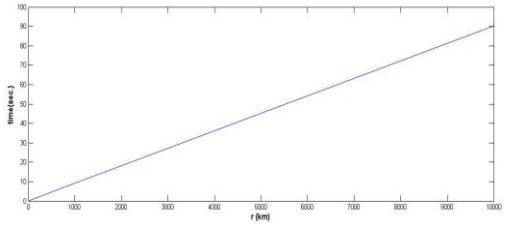

Fig. 4. encryption time

VII. CONCLUSION

Distance function evaluation over encrypted numbers is a challenging task. This paper presents a new method to address such a task. We have utilized the amazing properties of OPSE to design a secure protocol for evaluating the great circle distance between two encrypted coordinates. Interestingly, we have used the cloud servers to perform and store the collection of points and perform the distance evaluation without decryption. The practical value of our work came when a data owner uploads a set of secret locations into cloud server and allows the server to measure the distance of users without compromising the privacy of participant points. Several experiments have been conducted to investigate the performance of our work.

REFERENCES

[1] Y. Muto, K. Okano, S. Kusumoto, A visualization technique for unit testing and static checking with caller-callee relationships. *JoC J.*, 2, 1-8, 2011.

[2] X. Wang, Y. Sang, Y. Liu, Y. Luo, Considerations on security and trust measurement for virtualized environment. *JoC J.*, 2, 19-24, 2011.

[3] X. Xie, H. Jiang, H. Jin, W. Cao, P. Yuan, L. T. Yang, Metis: A profiling toolkit based on the virtualization of hardware performance counters. *HCIS J.*, 2, 2-15, 2012.

[4] G. Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*. O'Reilly, 2009.

[5] S. Subashini, V. Kavitha, A survey on security issues in service delivery models of cloud computing. *Network Computer Application J.*, 7, 1-11, 2011.

[6] R. Chow, P. Golle, , M. Jakobsson, E. Shi, J. Staddon, , R. Masuoka,., J. Molina Controlling data in the cloud: outsourcing computation without outsourcing control. *Proceedings of the 2009 ACM workshop on cloud computing security (CCSW '09)*, Chicago, Illinois, 9-13 November, 2009, pp. 85-90. ACM, New York, NY, USA.

[7] T. A Slocum, R. B. McMaster, F. C. Kessler, H. H. Howard, "Thematic Cartography and Geographic Visualization", Upper Saddle River: Pearson Prentice Hall, 2005.

[8] http://www.math.uwaterloo.ca/tsp/concorde/

[9] J. Cook, Williamm, G. Espinoza Daniel, Marcos Goycoolea., "Generalized Domino-Parity Inequalities for the Symmetric Traveling Salesman Problem", Mathematics of Operations Research 35:2, 479-493, 2010.

[10] D. L. Applegate, R. E. Bixby, V . Chvatal, W. Cook, , D.G. Espinoza, K. Helsgaun,., "Certification of an optimal TSP tour through 85,900 points". Operations Research Letters, 37,2009, pp. 11-15.

[11] A . Boldyreva, N. Chenette, Y . Lee, A. O. Neill, "Order-preserving symmetric encryption," In: Proceedings of Eurocrypt'09, vol. 5479, LNCS. Springer, 2009.

[12] A. Ibrahim, H. Jin, A. Ali Yassin , D. zou, , "Approximate Keyword-based Search over Encrypted Cloud Data", In: Proceedings of ICEBE, PP.238-245, 2012.

[13] G. T. McCaw,), "Long lines on the Earth", Empire Survey Review 1 (6): 259–263, 1932.