Computing the shortest route among multiple points without revealing their geographical locations

Ayad Ibrahim Abdulsada
Department of Computer Science,
College of Education for pure sciences, Basrah University, Basrah, 61004, Iraq.
Email: mraiadibraheem@yahoo.com

Ali Hussein Rason
Department of Mathematics,
College of Education for pure sciences,
Basrah University, Basrah, 61004, Iraq.
Email: aliresan@ymail.com

Ali A. Yassin
Department of Computer Science,
College of Education for pure sciences, Basrah University,
Basrah, 61004, Iraq.
Email: aliadel79yassin@gmail.com

Abstract

Travelling salesman problem (TSP) represents a classical optimization problem. Its main goal is to find the shortest route when visiting multiple points. However, the state of the art methods are generally assumed that the geographical locations are not private. This reduces the utilization of these methods to work within public environments. Essentially, this assumption limits more practical applications, e.g., the shortest route among military bases, where the geographical locations of such bases are confidential. This paper presents a method for privately computing the shortest route among multiple points on the Earth without compromising the privacy of their locations.

Key words: TSP, privacy preserving location services, great circle distance, genetic algorithms, cryptography.

1. Introduction

Travelling salesman problem (TSP) can be summaries as follows [1]: Given a set of *n* points, and the distance between each pair of points, the TSP goal is to find the shortest route that visits each point exactly once and returns to the starting point.

The TSP has several applications such as transportation and logistics applications, for example the problem of arranging school bus routes to pick up the children in a school district. Another example is to find shortest routes through selections of airports in the world. In these applications, the concept distance represents travelling times or cost.

Unfortunately, existing approaches of TSP suppose 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. Such a process is referred as *private traveling salesman problem* (PTSP). PTSP is formally described to find the permutation π of points that have the minimum cost.

$$PTSP(1,2,\ldots n) \rightarrow \pi(1,2,\ \ldots,n)$$

There are two basic approaches for solving TSP: *exact and approximate* approaches. The exact approach [2] uses the brute force search to try all permutations and decide which one is the optimal. Given *n* the number of points, the total number of possible routes is defined as *(n-1)!/2*. For large *n* value, such approach takes huge computational time. On the other hand, the approximate approach [3] never guarantees an optimal solution but gives near optimal solution in a reasonable computational time. Our work focuses on the approximate approach due to its efficiency.

Under the *geographic coordinate system* [4], 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 achieve PTSP is to utilize a *trusted third party* (TTP). Each point sends its location to TTP. Then the latter investigates and tells the entire points the shortest route among them. 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 address the problem of how to find the shortest route among a set of points on the Earth without revealing their geographical locations. We focus primarily on security, where protecting the coordinates is necessary. Specifically, our scheme uses the *genetic algorithm* (GA), a well-known optimization tool to select the shortest route, and employs the *order preserving symmetric encryption* (OPSE) to protect the confidentiality of coordinates.

The remainder of this paper is organized as follows. Section 2 discusses the related work. The problem definition is explained in Section 3. Section 4 introduces the preliminary techniques. We provide our proposed scheme in Section 5. Section 6 gives performance investigations. Finally, the whole paper is concluded in Section 7.

2. Related Works

The travelling salesman problem of visiting all 3,810 points in a circuit board was solved using *Concorde TSP Solver*[1]. The computation took approximately 15.7 CPU-years [5]. 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* [6].
Approximate approach can find solutions for millions of points within a reasonable running time. Lin–Kernighan is one of the best heuristics for solving the Euclidean travelling salesman problem. Briefly, it involves swapping pairs of sub-tours to make a new tour. It works by switching two or three paths to make the tour shorter [7]. Applying GA to the TSP is illustrated in many techniques [8, 9, 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,

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

on the other hand, measures the distance between two points along the surface of the Earth without compromising the privacy of their coordinates.

3. Problem Definition

Let $V = \{1, ..., n\}$ be a set of points, $A = \{(t, s):t, s\in V\}$ be the edge set, and $d_{ts}=d_{st}$ be a cost measure associated with edge $(t, s)\in A$. The TSP is the problem of finding a minimal length route that visits each point once. In this case, point $i\in V$ is given by its coordinates (φ_i, λ_i), and d_{ts} is the great-circle distance between the points t and s.

The optimization in our problem involves permutes the set of points in the right order π. So, the solution consists of a list of points in the order visited. The cost function for the permutation π is defined as:

$$\cos t = \sum_{i=1}^{n} d_{(\pi_{(i)}, \pi_{(i+1)})} \qquad \dots (2)$$

Where $\pi_{(i)}$ is the index of point at position i in the permutation π. In our work, we use the genetic algorithm to find the optimum permutation of the entire points.

Revealing the geographical coordinates of some secret locations, for example military bases, to other parties may jeopardize their privacy. Points usually have to encrypt their coordinates to combat unsolicited access. However, none of the existing encryption methods allow the user to evaluate the distance over the encrypted data. So, it is desirable to encrypt the coordinate values in such a way that allows evaluating their distance function without decryption. To do so, we employ the recently developed primitive OPSE to perform such a task. More details about this primitive are listed in Section 4.2.

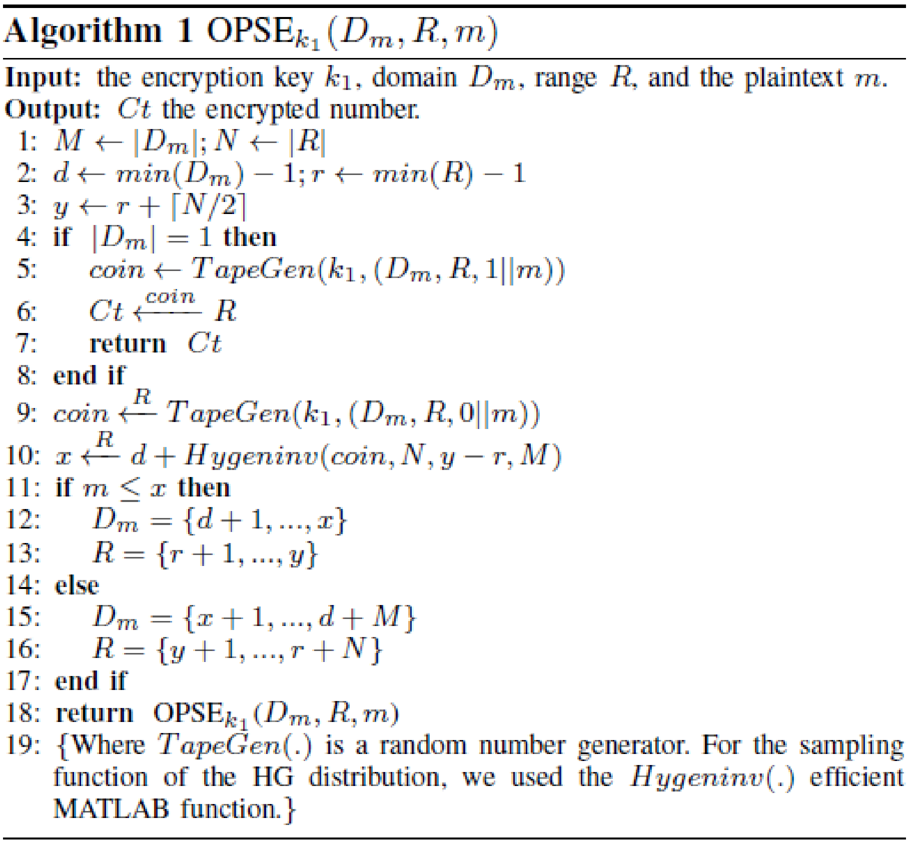
4. Preliminary Techniques

4.1 Genetic Algorithm

Genetic algorithm GA [11] is a method for solving both constrained and unconstrained optimization problems. Its work is based on natural selection, the process that drives biological evolution. A basic GA starts with a randomly generated population of individual solutions. The genetic algorithm repeatedly modifies the population. At each step, the genetic algorithm selects individuals at random from the current population to be parents. Parents are then mated to produce offspring and some go through a mutating process. Each individual has a fitness value telling us how good they are. Over successive generations, the population "evolves" toward an optimal solution. Applying GA to the TSP involves implementing a crossover routine, a measure of fitness, and also a mutation routine. A good measure of fitness is the actual length of the solution.

4.2 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 [12] 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 $D_m = \{1, . . . , M\}$ to range $R = \{1, \cdot\cdot\cdot ,N\}$, 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 [14]) represents the encryption function of OPSE scheme.



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

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
    $D_m = \{d + 1, ..., x\}$
    $R = \{r + 1, ..., y\}$
else
    $D_m = \{x + 1, ..., d + M\}$
    $R = \{y + 1, ..., r + N\}$
end if
return $\text{OPSE}_{k_1}(D_m, R, m)$
{Where $TapeGen(.)$ is a random number generator. For the sampling
function of the HG distribution, we used the $Hygeninv(.)$ efficient
MATLAB function.}
```

5. Proposed Scheme

The work of our proposed scheme is divided into three steps: coordinates mapping, coordinates encryption, and optimum permutation steps. In the first step, each point transforms its spherical coordinates into Cartesian ones. In the second step, the entire *n* points encrypt their coordinate's values before revealing them. The third step computes the shortest route over to the encrypted coordinates.

5.1 Coordinates Mapping

Assume that the Earth is a sphere of radius *r*. Each point *i=1,…,n* converts its spherical coordinates (φ_i, λ_i) from degree into radian units by multiplying each coordinate with the constant *pi*/180. Then it transforms them into three-dimensional space as follows [4]:

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

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.

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

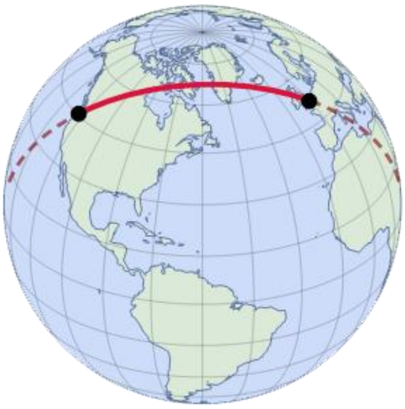


Figure (1): Great circle distance

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

5.2 Coordinates Encryption

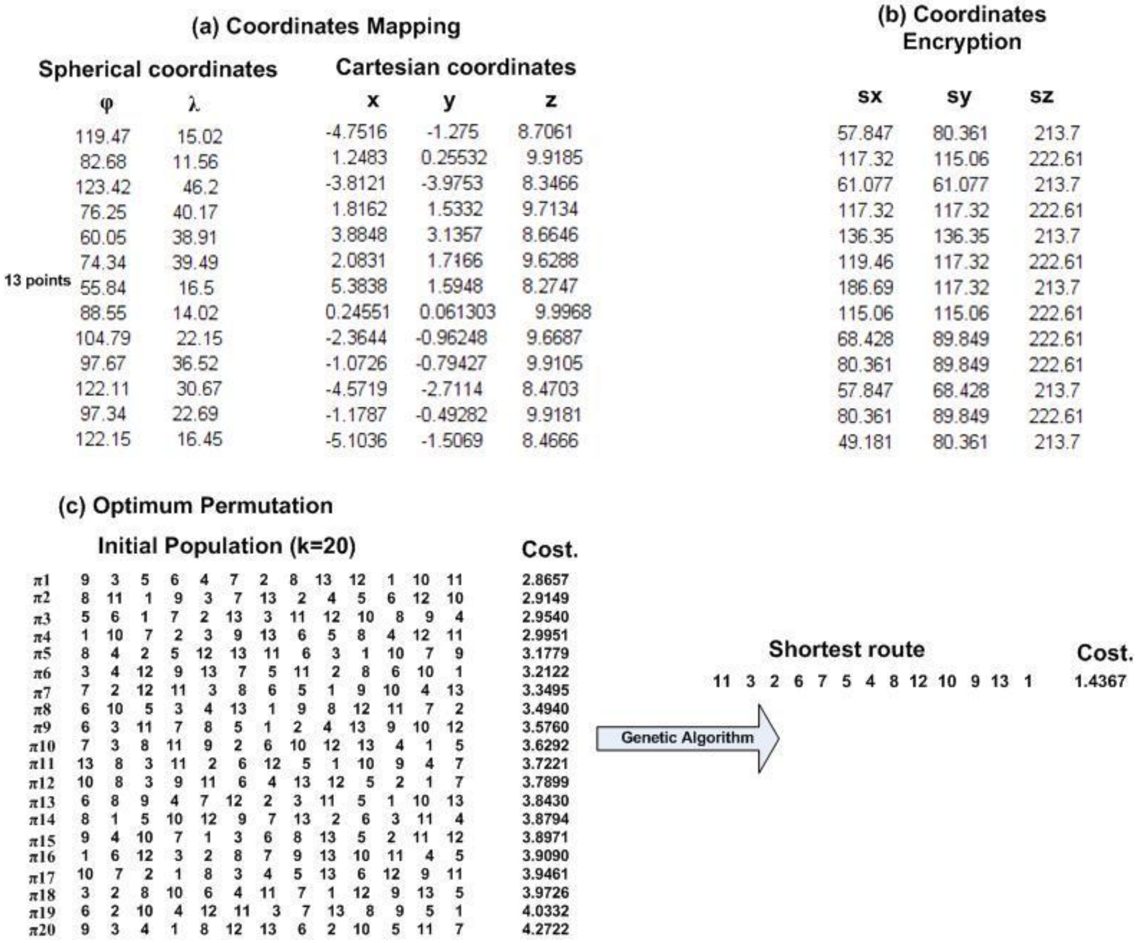
This step encrypts the Cartesian coordinates *(x,y,z)* of each point. As explained before OPSE preserves the numerical ordering of its original numbers. Such an amazing properly enables us to conduct the distance function over the encrypted numbers without decryption. Initially, the participant points *V={1…n}* agree on the initial parameters of the OPSE, which are: secret key $k_{1,}$ domain D_m, and range R. Then each point i encrypts its 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 $(sx_i,\ sy_i,\ sz_i),\ i=1,...,\ n$. 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

(c) Optimum Permutation

	Initial Population (k=20)													Cost.
π1	9	3	5	6	4	7	2	8	13	12	1	10	11	2.8657
π2	8	11	1	9	3	7	13	2	4	5	6	12	10	2.9149
π3	5	6	1	7	2	13	3	11	12	10	8	9	4	2.9540
π4	1	10	7	2	3	9	13	6	5	8	4	12	11	2.9951
π5	8	4	2	5	12	13	11	6	3	1	10	7	9	3.1779
π6	3	4	12	9	13	7	5	11	2	8	6	10	1	3.2122
π7	7	2	12	11	3	8	6	5	1	9	10	4	13	3.3495
π8	6	10	5	3	4	13	1	9	8	12	11	7	2	3.4940
π9	6	3	11	7	8	5	1	2	4	13	9	10	12	3.5760
π10	7	3	8	11	9	2	6	10	12	13	4	1	5	3.6292
π11	13	8	3	11	2	6	12	5	1	10	9	4	7	3.7221
π12	10	8	3	9	11	6	4	13	12	5	2	1	7	3.7899
π13	6	8	9	4	7	12	2	3	11	5	1	10	13	3.8430
π14	8	1	5	10	12	9	7	13	2	6	3	11	4	3.8794
π15	9	4	10	7	1	3	6	8	13	5	2	11	12	3.8971
π16	1	6	12	3	2	8	7	9	13	10	11	4	5	3.9090
π17	10	7	2	1	8	3	4	5	13	6	12	9	11	3.9461
π18	3	2	8	10	6	4	11	7	1	12	9	13	5	3.9726
π19	6	2	10	4	12	11	3	7	13	8	9	5	1	4.0332
π20	9	3	4	1	8	12	13	6	2	10	5	11	7	4.2722

Shortest route													Cost.
11	3	2	6	7	5	4	8	12	10	9	13	1	1.4367

Figure (2): Basic steps of the proposed scheme

5.2 Optimum Permutation.

As mentioned before, the solution of our optimization problem can be viewed as finding the optimum permutation of the points' indexes. Such that, visiting the entire points, according to this permutation, find the route of the minimum distance cost. Genetic algorithm solve such an optimization problem as follows: first, it begins by creating a k random initial population of random permutations $(\pi 1, \pi 2, ..., \pi k)$. Each permutation π is of length n, where n is the number of the involved points. Second, the algorithm then creates a sequence of new populations. At each iteration, the algorithm uses the individuals in the current generation to create the next population. To do so, the algorithm performs the following steps [14]:

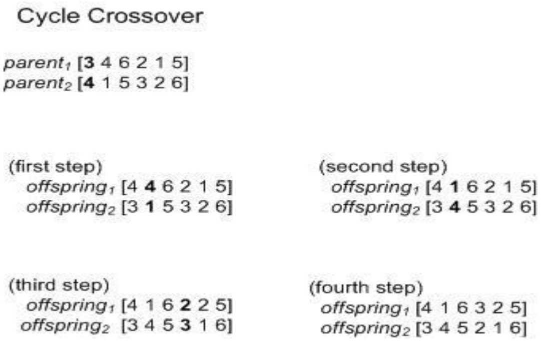
1. Scores each permutation πi, $i=1,\dots,k$ of the current population by computing its fitness value as in equation 2.
2. Scales the raw fitness scores to convert them into a more usable range of values.
3. Selects members, called *parents*, based on their fitness.
4. Some of the individuals in the current population that have lower fitness are chosen as *elite*. These elite individuals are passed to the next population.
5. Produces children from the parents. Children are produced either by making random changes to a single parent—*mutation*—or by combining the vector entries of a pair of parents—*crossover*.
6. Replaces the current population with the children to form the next generation.

The algorithm stops when one of the stopping criteria is met. The fitness function of the permutation π is explained as in equation 2. But here we use the distance d_{sr} to measure the distance over the encrypted coordinates of points t and s as follows:

$$d_{ts} = \sqrt{(sx_t - sx_s)^2 + (sy_t - sy_s)^2 + (sz_t - sz_s)^2} \quad \dots(3)$$

The crossover operator is the *cycle crossover* (CX) [14]. Given two parents $parant_1$ and $parent_2$, we randomly select a location and the integers are exchanged between the two parents. Unless the exchanged integers are the same, each offspring has a duplicate integer. Next the repeated integer in $offspring_1$ is switched with the integer at that site in $offspring_2$. Now a different integer is duplicated, so the process iterates until we return to the first exchanged site. At this point each offspring contains exactly one copy of each integer from *1* to *n*.

Consider two parents of length 6:



Cycle Crossover

parent$_1$ [**3** 4 6 2 1 5]
parent$_2$ [**4** 1 5 3 2 6]

(first step)
offspring$_1$ [4 **4** 6 2 1 5]
offspring$_2$ [3 **1** 5 3 2 6]

(second step)
offspring$_1$ [4 **1** 6 2 1 5]
offspring$_2$ [3 **4** 5 3 2 6]

(third step)
offspring$_1$ [4 1 6 **2** 2 5]
offspring$_2$ [3 4 5 **3** 1 6]

(fourth step)
offspring$_1$ [4 1 6 3 2 5]
offspring$_2$ [3 4 5 2 1 6]

At this point we have exchanged the 2 for the 3 and there are no repeated integers in either string, so the crossover is complete.

The mutation operator randomly chooses a string, selecting two random sites within that string, and exchanges the integers at those sites. Fig.2c explains an example of how to use the genetic algorithm to return the optimum route covering 13 points.

6. 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=6371km [15]. The domain of OPSE D_m= [-$r*10$ ..... $r*10$] and the range R=[-$r*100$ ... $r*100$]. For genetic algorithm, we used the proportional selection and Roulette wheel sampling method to select the parents. The stopping condition depends on identifying a maximum number of generations *Maxit*. Table 1 shows the common symbols used in our experiments.

Table 1: Symbols used in our experiments

Symbol	Meaning
n	number of participant points
r	radius of the Earth
k	population size
x,y,z	Cartesian coordinates
sx, sy, sz	encrypted coordinates
D_m	domain of OPSE
R	range of OPSE
cost	distance cost of the route covering *n* points
Maxit	No. of maximum generations.

6.1 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.

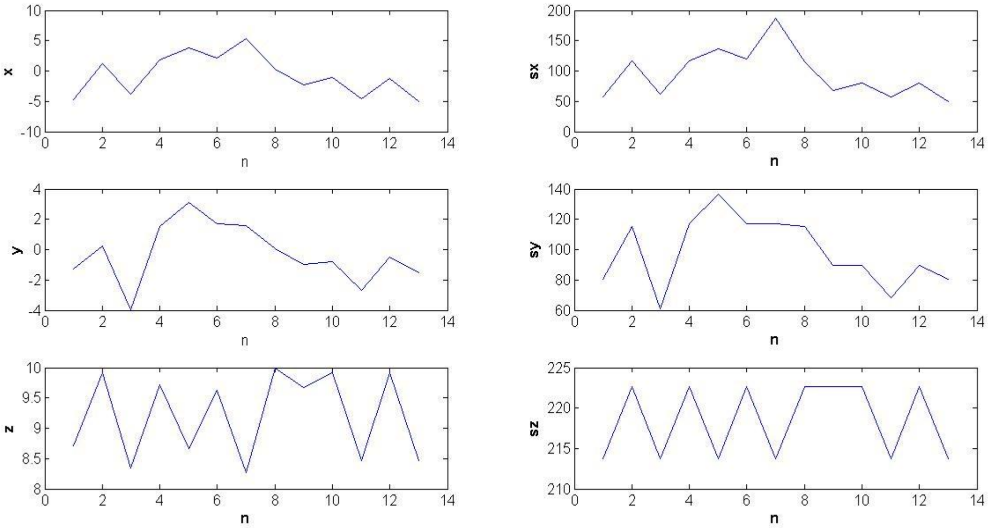


Figure (3): correctness of OPSE

6.2 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 iterations to get the last window in OPSE.

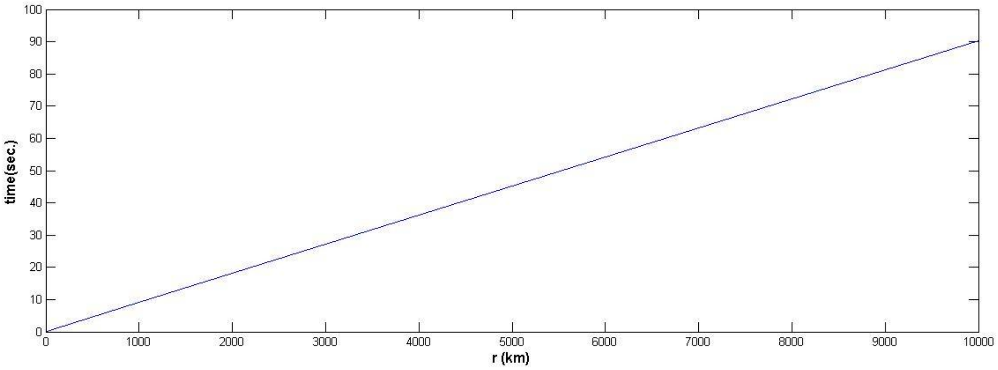


Figure (4): encryption time

6.3 Genetic Performance

This experiment shows the effect of genetic algorithm iterations *Maxit* on the cost of the optimal route. Fig. 5 shows two cases: *minimum* and *average* costs. The first one represents the cost of the better route in the current population. The second case represents the average cost of the entire population. Both cases are drowning as increasing the generation number. It is easy to see that more iterations result a route of low cost.

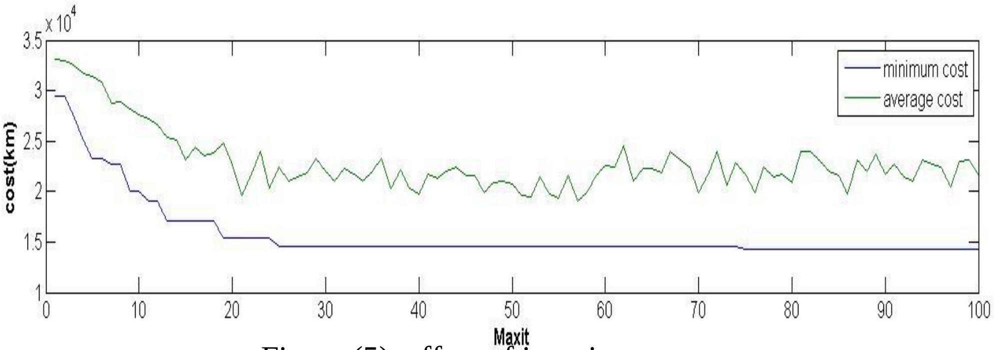


Figure (5): effect of iterations on route cost

7. Conclusion

Evaluation distance function over encrypted numbers is a challenging task. This paper presents a new method to find the shortest route among multiple points without compromising the privacy of their geographical locations. We have utilized the amazing properties of OPSE to design a secure protocol for evaluating the great circle distance. Interestingly, we have used the genetic algorithm to find the optimal permutation to the indexes of cites according to the fitness of encrypted data. The practical value of our work came when multiple parties, such as military bases need to find the shortest path that covers them without revealing their geographical locations. Several experiments have been conducted to investigate the performance of our work.

References

[1] Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. (2006), "*The Travelling Salesman Problem*", ISBN 0-691-12993-2.

[2] Woeginger, G.J. (2003), "*Exact Algorithms for NP-Hard Problems: A Survey*", Combinatorial Optimization – Eureka, You Shrink! Lecture notes in computer science, vol. 2570, Springer, pp. 185–207.

[3] Li, K.; Kang, K.; Zhang, W.; Li, B (2008), "*Comparative analysis of genetic algorithm and ant colony algorithm on solving travelling salesman problem*", In: Proceedings of the IEEE International Workshop on Semantic Computing and Systems, pp.72-75.

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

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

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

[7] Helsgaun, K. (2009), "*General k-opt submoves for the Lin-Kernighan TSP heuristic*", Mathematical Programming Computation.

[8] Johnson, D.S.; McGeoch, L.A. (2002), "*Experimental Analysis of Heuristics for the STSP, The Travelling Salesman Problem and its Variations*", Gutin & Punnen (eds), Kluwer Academic Publishers, pp. 369-443.

[9] Hongwei, D.; Yu, Y.; Cunhua, Li. (2011), "*Stochastic Compact Quantum Crossover Based Clonal Selection Algorithm for Travelling Salesman Problems*", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, No. 9, pp. 371 -378.

[10] Albayrak, M.; Allahverdi, N. (2011), "*Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms*", Expert Systems with Applications, vol. 38, no. 3, pp. 1313–1320.

[11] Chatterjee, S.; Carrera, C.; Lynch, L. A. (1996), "*Genetic algorithms and traveling salesman problems*", European Journal of Operational Research, vol. 93, no. 3, pp. 490-510.

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

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

[14] Engelbrecht, A. (2007), "*Computational Intelligence - An Introduction*", second edition, Wiley.
[15] McCaw, G. T. (1932), "*Long lines on the Earth*", Empire Survey Review 1 (6): 259–263.