Outsource-Secured Calculation of Closest Pair of Points

Chandrasekhar Kuruba, Kethzi Gilbert, Prabhav Sidhaye, Gaurav Pareek(✉), and Purushothama Byrapura Rangappa

National Institute of Technology, Ponda, Goa, India
chandusree928@gmail.com, kethzi.gildona@gmail.com, prabhavsid995@gmail.com, {gpareek,puru}@nitgoa.ac.in

Abstract. Outsourcing data/computation intensive tasks to servers having great computing power and data analytics skills is gaining popularity. While this outsourcing model, due to its cost efficiency, has been widely used by numerous clients, making sure that loss of privacy and integrity of results are not affected remain as challenges, especially in public cloud infrastructure. For addressing these challenges, clients must outsource their data in a privacy-preserving and verifiable manner. The cost of assuring both privacy of data and correctness of results must impose cost marginally less than the cost of actual computation. In this paper, we address the problem of secure outsourcing of Closest Pair of Points computation. Finding Closest Pair of Points is central to many complex applications like clustering. Our scheme involves the client sending encrypted points to the server and receiving the result which is a pair of points (with smallest distance between them) along with a proof of correctness. Data encryption done to ensure privacy of input points must be such that the encrypted points retain the same order as the original points. For this, we designed and used a novel encryption scheme which is additively homomorphic and order-preserving for encrypting input points in our protocol. The protocol requires the server to compute almost all distances to be able to provide the proof of it having computed the results honestly.

Keywords: Verifiable computing · Secure outsourcing · Closest-pair-of points · Cloud security

1 Introduction

In this age of distributed computing devices, cloud computing, more precisely, XaaS (Anything as a Service) is gaining wide acceptance. As a result to this, notion of public clouds has also come into existence. Servers offer services in terms of heavy computations (like complex data mining operations) on big data sets and are paid by the clients which are limited by the computing resources and/or memory storage, for the same. Clients who use any cloud services rely heavily on the fact that the cloud server is not interested in cheating on the computations or breaching the privacy of the client's data. But the client can return results which look like the results client is expecting but the server may not be

P. Mueller et al. (Eds.): SSCC 2016, CCIS 625, pp. 377–389, 2016.
DOI: 10.1007/978-981-10-2738-3_33

calculating them honestly and returning random values in a particular range, for example. In such cases, on one hand, integrity and privacy of client's data is compromised and on the other hand, server saves its computation efforts (in terms of CPU cycles) and memory resources, which is a motivation good enough for server to "cheat". There is motivation for the server to not care about the client's data privacy also. Consider the case where the server which is given the log of web pages visited by a set of users for performing complex data mining operations, is contacted by an advertising agency looking for such logs for user's behavioral analysis. The server can gain significantly by disclosing the logs to the agency. Considering many critical applications for which the data/computations are increasingly being outsourced, one can imagine the threat to his data privacy and reputation. The above sets motivation for secure outsourcing of data and computations. The goals of a secure-outsourcing protocol include securing input and output from disclosure, cheating-proof execution of computations and limited (extra) work on the client's side. A variety of problems have been securely outsourced under two-server model and single server model. Verifiable computing deals with checking the authenticity of the results returned from the server. Effectiveness of a verifiable scheme is the average fraction of the total work a malicious server has to do in order to successfully cheat the client despite the verification algorithm being part of the scheme. This fraction should be very close to 1 for a good verifiable scheme.

In this paper, we consider one of the fundamental geometric problems, computing the closest-pair-of points. Closest Pair of Points deals with identification of the pair with minimum Euclidean distance amongst a given set of points. Since the problem has repeated computation of Euclidean distances between different pair of points, it serves as a primitive step for many data mining applications. The client sends the set of points and the server should compute distance between each pair of points and return the pair of points with minimum distance as output. Secure outsourcing of such a computation requires that the points are encrypted and the server provides the proof of correctness along with the result pair. The encryption we used for encrypting the points is order-preserving so that the relative order of computed distances between the encrypted points is same as for the original points.

Our contributions can be summarized as follows. We addressed the problem of secure outsourcing of computation of Closest Pair of points. Using the proposed scheme, a client delegates the computation in privacy preserving and verifiable manner. The work done by the client for preprocessing the input points and processing the result returned by the server for obtaining the final result amounts to no more than $O(n)$ where n is the number of input points. For achieving this, we have designed and used a novel order-preserving encryption scheme for encryption of input points. A theoretical analysis of correctness and security of the result verification protocol is also presented. We have also proposed a new algorithm for computing Closest Pair of Points which outperforms both brute-force ($O(n^2)$) and divide-and-conquer ($O(nlogn)$) versions of it.

The rest of the paper is organized as follows. Section 2 presents review of important existing literature on privacy preserving and verifiable computation

outsourcing. Section 3 covers all the preliminaries required for understanding a secure outsourcing protocol for any important computation. In Sect. 4, complete description of the proposed scheme for outsourcing closest pair of points is presented which includes proposed order-preserving encryption scheme in Sect. 4.1, outsourcing algorithm in Sect. 4.2, discussion on adaptibility of the scheme to the two-server model in Sect. 4.3. Randomized verification for the outsourcing scheme is presented in Sect. 4.4 along with its analysis in Sect. 4.5. A new algorithm for computing closest pair of points that outperforms the existing algorithms existing ones is proposed in Sect. 4.6 along with the experimental results in Sect. 4.7.

2 Related Work

Atallah et al. [8] propose the notion of disguise that can be introduced by the client during the preprocessing phase of a outsource-secured computation. Disguising the computations hides the computation along with the input and output data from the server. Random object generation is considered important in disguising. So, the security and effectiveness of all these disguising schemes rely on the assumption that random component is generated using a "good enough". The work also encompasses techniques that can be applied for domain and dimension modification.

Certain classes of algebraic and differential computations have also been considered as candidates for secure outsourcing [9]. It is shown that approximating many of the algebraic differential equations can be reduced to solving classical problems like Abstract Equation(AE), Boundary Value Problem(BVP), Initial Value Problem (IVP) with secret parameter etc. Also, non-linear equations can be reduced to linear forms. Next we present the classes of problems and their reduction to the "simpler" problems as in [9] along with basic methodology used for securely outsourcing them.

Hohenberger et al. [10] formally define the notion of outsource security in presence of untrusted server(s). The notion of α-efficient and β-checkable Secure Outsourcing algorithms is also given after which a $(O(\frac{\log^2 n}{n}), \frac{1}{2})$-outsource secure implementation of modular exponentiation is presented under the two untrusted program model.

Secure outsourcing of product of two large matrices in two server setting appeared in 2008 [1]. The extension for cheating detection for the protocol depends on the infeasibility of predicting a matrix A given a the matrix AR or RA (but not both at the same time), where R is a radom matrix.

Linear Algebra Computations serve as building block for many cryptographic algorithms. Problem of secured outsourcing of characteristic polynomial and eigenvalues of a matrix is addressed by Hu et al. [12] without any cryptographic assumptions. The protocol is non-interactive. Verification of result requires $O(n^2)$ local computations by the client. The protocol is efficient with client needing only $O(n^2)$ multiplications to calculate the characteristic polynomial and eigenvalues. Mohassel et al. in [13] show that if there exists a homomorphic

encryption scheme that is distinctive, associative, additive and multiplicative then the matrix computations can be securely outsourced with client doing at most $O(n^2 logn)$ work.

To combat limited computing power of client, Wang et al. [14] developed a mechanism for securely outsourcing large-scale system of linear equations along with a robust cheating detection mechanism. One-time setup phase of the protocol takes $O(n^2)$ cost. Finally, $O(n)$ cost is incurred by client to calculate the n variables in the problem. When a key (r) of size 768-bit, the protocol performs well above the computation baseline for the problem. However, using the 1024-bit key leads the protocol to perform badly as compared to what is considered as computation baseline for the LE problem.

Problem of public verification of results for outsourced computations without the verifier needing any key [15] appeared in the year 2012. New protocols to outsource-secured *evaluation of high degree polynomials* and *matrix multiplication* are proposed under stronger adversarial models.

The problem of privacy preserving distributed k-means clustering of arbitrary partitioned data was first addressed by Geetha et al. [16] after which improved versions of outsource-secure k-means clustering appeared in [17,18]. Bunn et al. [17] proposed a slightly more efficient private multi-party k-means clustering algorithm. Additionally, it also addresses the problem that could arise from dishonest contributing sites. The scheme works under the assumption of existence of semantically secure Homomorphic encryption scheme.

3 Preliminaries

3.1 Outsourcing Models

Outsourcing is a method by which a weak client asks the server to perform computations which cannot be done by the client. In *single-server model*, there is one server and one client. The server does the entire computation by itself. The data on which the computation is to be done is passed on by the client to the server. Since the server does the entire computation by itself it also has the result of the desired computation. In case of an untrusted server, the client has high risk of losing his data privacy and/or integrity of computed output. Whereas in *two-server model*, the computation is done in parts by the two servers present and the partial results are put together either by one of the servers or by the client [1]. We follow a model where the servers are allowed to communicate with one another. However, there exist outsourcing models where there exists no communication between the servers. In our algorithm, we assume that the two servers do not collude, hence there is no need for encryption. The claim that the solution proposed in this paper fits both the models is proved in Sect. 4.3.

3.2 Requirements for Our Encryption Scheme

To ensure data privacy, the input points need to be encrypted before sending to the server. The encryption algorithm we propose for securely outsourcing closest pair of points is order preserving and homomorphic.

Order Preservation. Since we have to identify the closest pair of points using distances the encryption must not change the order of the original points i.e. the relative order of the points must not be disturbed.

For instance, consider two points P_1 and P_2. Also consider an encryption scheme with the encryption algorithm E. Suppose $P_1 > P_2$ then $E(P_1) > E(P_2)$ where $E(P_1)$ and $E(P_2)$ are encrypted values of P_1 and P_2 respectively.

Homomorphic Property. The encryption scheme we use must have homomorphic property over addition. The encryption scheme is said to be additively homomorphic if decrypting the sum of encrypted values will give us the sum of plaintext values. Consider an encryption scheme which gives the encrypted values of P_1 and P_2 denoted by $E(P_1)$ and $E(P_2)$. An encryption scheme is additively homomorphic if

$$Dec(E(P_1) + E(P_2)) = P_1 + P_2$$

Where $Dec(E(P_x))$ is the decryption of $E(P_x)$.

4 Outsourcing of Closest Pair of Points

Outsourcing of the closest pair of points problem requires that a client provides a server a set of n points and the server computes and returns a pair with minimum distance. We discuss all the steps of the proposed scheme in detail and verify the claims we provide with each step.

4.1 The Proposed Order Preserving Symmetric Encryption Scheme

Order preserving encryption scheme is an encryption scheme that preserves the numerical ordering of its input. It was originally developed for enabling efficient range queries over encrypted database [2]. The construction of this scheme is based on the uncovered relation between the random order preserving function and hyper-geometric probability distribution. The order preserving function is a one-one function from the domain set of cardinality M and Range set of cardinality N such that $N > M$. The function can be defined as selection of M out of N ordered items and simultaneously fulfilling the requirement of a good encryption scheme - "as random as possible". We have implemented this scheme for sample data points shown in Fig. 1a and the data points after encryption are shown in Fig. 1b. It can be seen from the output of encryption the data points get reflected about the line $x = x_{max}/2$ and $y = y_{max}/2$.

Assumptions. Without loss of generality, we can make the following assumptions on the data points:

1. All the data points lie in the first quadrant of the co-ordinate system i.e. for all points $P(x, y)$ both x and y are greater than 0.

2. None of the data points lie on either x or y axis i.e. no points have either abscissa or ordinate as 0.

Key Generation. We begin by identifying the maximum value that the coordinates of our data points take. Let this maximum value be P.

$$P = Max(x_1, x_2, x_3, \ldots, x_n, y_1, y_2, y_3, \ldots, y_n)$$

Randomly choose a number r such that $r \geq 3$ ($r \in \mathbb{Z}_m$) and a random $R \in [1, P-1]$. Let $Q = r \times P$ and for every point $P_i = (x_i, y_i)$, compute $x_{inorm} = x_i \times R/P$ and $y_{inorm} = y_i \times R/P$

Encryption. Now we encrypt a point $P_i(x_i, y_i)$ as

$$Enc(x_i) = Ex_i = Q - P - x_i - x_{inorm}$$
$$Enc(y_i) = Ey_i = Q - P - y_i - y_{inorm}$$

where $Enc(a)$=Encryption of data value a. Therefore, the encrypted value of point P_i is

$$Enc(P_i(x_i, y_i)) = EP_i(Ex_i, Ey_i)$$

Claim. Euclidean distance computation can be directly performed on EP_i and EP_j i.e. $ED(EP_i, EP_j) = (1 + R/P) \times \ ED(P_i, P_j)$, where ED is the Euclidean distance between points P_i, P_j

Proof.

$$\begin{aligned} E_{x_i} &= Q - P - x_i - x_{inorm} \\ &= r \times P - P - x_i - R/P \times x_i \\ &= P(r-1) - x_i((P+R)/P) \end{aligned}$$

Similarly,

$$\begin{aligned} E_{y_i} &= P(r-1) - y_i((P+R)/P) \\ D(EP_i, EP_j) &= [(E_{x_i} - E_{x_j})^2 + (E_{y_i} - E_{y_j})^2]^{1/2} \\ &= (P+R)/P \times D(P_i, P_j) \end{aligned}$$

Lemma 1. *If* $D(P_i, P_j) > D(P_x, P_y)$ *then* $ED(EP_i, EP_j) > ED(EP_x, EP_y)$

Consider,

$$D(P_i, P_j) > D(P_x, P_y) \implies (1 + R/P)D(P_i, P_j) > (1 + R/P)D(P_i, P_j) \quad (1)$$

Since R,P are all positive, Using property proved above on Eq. 1 we get, $ED(P_i, P_j) > ED(P_x, P_y)$

Because of the above property we can directly compare the Euclidean distances between all points and identify the closest pair. Our encryption scheme successfully hides actual distances between every pair of points.

4.2 Outsourcing Algorithm

Client

1 Has set of data points $P = \{P_1(x_1, y_1), P_2(x_2, y_2), \ldots, P_n(x_n, y_n)\}$.
2 Encrypts data points in P with an appropriate scheme to obtain the set $EP = \{EP_1, EP_2, \ldots, EP_n\}$
such that
$EP_i = Enc(P_i) = (Enc(x_i), Enc(y_i))$
3 Receives the encrypted closest pair of points from the server EP_x and EP_y
4 Decrypts the received pair of points to obtain the required results
$Dec(EP_x) = Dec(Enc(P_x)) = P_x$
$Dec(EP_y) = Dec(Enc(P_y)) = P_y$
P_x and P_y are the closest pair of points.

Server

1 Receives encrypted set of points $EP = \{EP_1, EP_2, \ldots, EP_n\}$ such that $EP_i = Enc(P_i) = (Enc(x_i), Enc(y_i))$
2 Computes closest pair of points EP_x, EP_y by computing Euclidean distances between encrypted points in set EP using any of the existing algorithms.
3 Sends the results EP_x, EP_y obtained in step 2 to the client.

Next, we show that the outsourcing algorithm we presented here fits the popular two-server model.

4.3 Two-Server Model

In this model, we divide the coordinates between two servers such that the sum of coordinates with the two servers gives the original coordinates. Each server computes the partial results and one of the servers finally combines them and returns them to the client. In this case, if servers are not assumed to collude, the data encryption is not a necessity because each server only has a part of the sensitive information. In our model, we use the following algorithm:

1. The client divides the points into two parts such that The client randomly subtracts the number (x_{i_1}, y_{i_1}) from the co-ordinates of points $P_i = (x_i, y_i)$ i.e.

$$P_{i_2} = (x_i - x_{i_1}, y_i - y_{i_1})$$

let $x_i - x_{i_1} = x_{i_2}$ and $y_i - y_{i_1} = y_{i_2}$
Therefore,

$$x_{i_1} + x_{i_2} = x_i$$
$$y_{i_1} + y_{i_2} = y_i$$

2. Let $P_{i_1} = (x_{i_1}, y_{i_1})$ and $P_{i_2} = (x_{i_2}, y_{i_2})$ Similarly, $P_{j_1} = (x_{j_1}, y_{j_1})$ and $P_{j_2} = (x_{j_2}, y_{j_2})$

3. P_{i_1}, P_{j_1} is sent to S_1 for all $i, j < n$ and P_{i_2}, P_{j_2} is sent to S_2 for all $i, j < n$
4. S_1 computes $D(P_{i_1}, P_{j_1})$. During this computation S_1 computes $(x_{i_1} - x_{j_1})$ and $(y_{i_1} - y_{j_1})$.
5. S_2 computes $D(P_{i_2}, P_{j_2})$. During this computation S_2 computes $(x_{i_2} - x_{j_2})$ and $(y_{i_2} - y_{j_2})$.
6. S_2 sends $(x_{i_2} - x_{j_2})$ and $(y_{i_1} - y_{i_2})$ as well as $D(P_{i_2}, P_{j_2})$ to S_1
7. S_1 combines the results it has computed and the data it has received to compute $D(P_i, P_j)$

Claim. Distance between P_i and $P_j = D(P_i, P_j)$ can be computed if first Server i.e. S_1 receives $D(P_{i_2}, D_{j_2})$ and $(x_{i_2} - x_{j_2}), (y_{i_2} - y_{j_2})$ from S_2

Proof. S_1 already computes

$$D^2(P_{i_1}, P_{j_1}) = (x_{i_1} - x_{j_1})^2 + (y_{i_1} - y_{j_1})^2 and (x_{i_1} - x_{j_1}) \text{ and } (y_{i_1} - y_{j_1})$$

S_2 sends

$$D^2(P_{i_2}, P_{j_2}) = (x_{i_2} - x_{j_2})^2 + (y_{i_2} - y_{j_2})^2$$
$$\text{and } (x_{i_2} - x_{j_2} \text{ and } (y_{i_2} - y_{j_2})$$

Adding information from S_1 and S_2,

$$(x_{i_1} - x_{j_1})^2 + (y_{i_1} - y_{j_1})^2 + (x_{i_2} - x_{j_2})^2 + (y_{i_2} - y_{j_2})^2 \tag{2}$$

Since we already have $(x_{i_1} - x_{j_1})$, $(y_{i_1} - y_{j_1})$, $(x_{i_2} - x_{j_2})$ and $(y_{i_2} - y_{j_2})$, computing $2(x_{i_1} - x_{j_1})(x_{i_2} - x_{j_2}) + 2(y_{i_1} - y_{j_1})(y_{i_2} - y_{j_2})$ and adding with Eq. 2,

$$(x_{i_1} - x_{j_1})^2 + (y_{i_1} - y_{j_1})^2 + (x_{i_2} - x_{j_2})^2 + (y_{i_2} - y_{j_2})^2 + 2(x_{i_1} - x_{j_1})(x_{i_2} - x_{j_2}) + 2(y_{i_1} - y_{j_1})(y_{i_2} - y_{j_2})$$

$$= [x_i - x_j]^2 + [y_i - y_j]^2$$
$$= D^2(P_i, P_j)$$

□

4.4 Randomized Verification Scheme

Along with privacy of actual input and output points, verifiability of the results returned by the untrusted server must also be ensured. This verification algorithm has mainly two phases. In the first phase, client pre-processes followed by outsourcing of input data along with a challenge. Second phase involves the server that computes the result and responds to the client with the output along with the response to the challenge. Response sent by the server becomes the basis for accepting or rejecting the result. The algorithm is as follows:

Input: Set of points $P = \{P_1, P_2, P_3, \ldots, P_n\}$ where $P_i = (x_i, y_i)$.

Output: (P_i, P_j) such that $\min\{d(P_i, P_j)\}$ for all $i, j \leq n$ and $\delta = (d(P_i, P_j))$.

Assumption: The distances between all pair of points are distinct.

Pre-process and Outsource

1. Client C randomly chooses $i, j \in \{1, \ldots, n\}$
2. Compute Euclidean distance between P_i, P_j call it α.
3. Send $P = \{P_1, P_2, \ldots, P_n\}$ and α to server S.
4. C removes P from storage and only stores (P_i, P_j).

Compute and Respond

1. Server computes Euclidean distance between $P_i, P_j \forall i, j \in \{1, \ldots, n\}$
2. Finds the closest pair (P_i, P_j) and distance between them. Call it d_{min}.
3. Finds x, y such that $\alpha = d(P_x, P_y)$
4. Sends $\pi = (d_{min}, (P_i, P_j), (x, y))$

Verification$(\pi, (i, j))$

1. Check if$(i = x$ and $j = y)$ true
   then accepts d_{min} as minimum distance
2. else rejects d_{min}

4.5 Analysis of the Verification Scheme

A result verification scheme for privately outsourced closest pair of points computation has been provided in the previous section. Verification of results returned by the server involves challenging the server with a distance between any randomly chosen pair of points. The server has to return the result of the desired computations (i.e. closest pair) along with the response corresponding to the challenge. In order to still successfully cheat, it has to find the pair of points the client used for computing the challenge. This method can be analyzed by calculating the average amount of work (Euclidean distance computations) the server has to do in order to find out the pair of points used for computing the challenge. Once the server computes response to the challenge, there is no motivation left for the cheating server to calculate the rest of distances and it returns arbitrary result (pair of points) to the client. Let there be n points, then the probability that server guesses the pair of points in one attempt (i.e. by calculating only one distance) is:

$$P(1) = \frac{1}{\binom{n}{2}}$$

If the server continues working out distances in whatever order it wishes, the probability of it succeeding after computing "w" distances is:

$$P(w) = \frac{\binom{1}{1} \times \binom{\binom{n}{2}-1}{w-1}}{\binom{\binom{n}{2}}{w}} = \frac{w}{\binom{n}{2}} \tag{3}$$

So, expected amount of work (Euclidean distance computations) done by the server in before it correctly obtains the challenge points:

$$E[W] = \sum_{w=1}^{\binom{n}{2}} w \times P(w) = \frac{(\binom{n}{2}+1)(2\binom{n}{2}+1)}{6}$$

The server can follow another rational approach by which it computes a fixed number of distances (say w') an upon failure in finding the challenge points, guesses from the remaining. In such a case, probability of success for the attacker is:

$$P'(w') = \frac{\binom{1}{1} \times \binom{\binom{n}{2}-1}{w'-1}}{\binom{\binom{n}{2}}{w'}} + (1 - \frac{\binom{1}{1} \times \binom{\binom{n}{2}-1}{w'-1}}{\binom{\binom{n}{2}}{w'}}) \times (\frac{1}{\binom{n}{2} - w'}) = \frac{w'+1}{\binom{n}{2}} \quad (4)$$

Similarly, average number of distance computations the server has to do in order to find the challenge is:

$$E'[W'] = \sum_{w'=1}^{\binom{n}{2}} w' \times P'(w') = \frac{(\binom{n}{2}+1)(\binom{n}{2}+2)}{3}$$

It is clear from (4) that this strategy gives the attacker a very limited advantage as compared to (3). Next, we discuss an algorithm for calculating the closest pair of points that performs better than both brute force and divide and conquer algorithm.

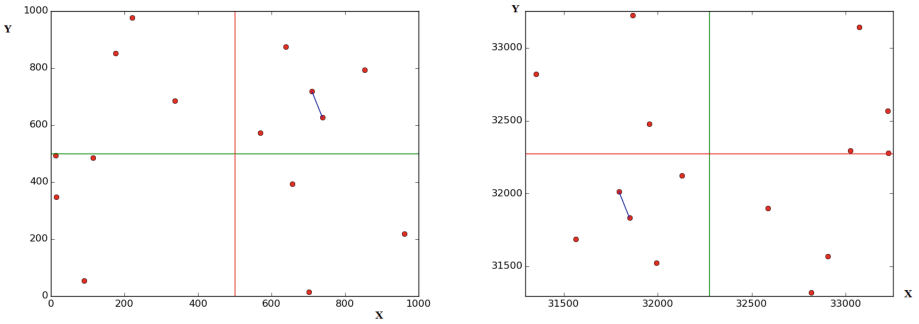
4.6 Proposed Algorithm for Computing Closest Pair of Points

We propose an algorithm to compute closest pair from a given set of points. In this algorithm we compute relative distances between points. We follow the principle that two points with close relative distance will be close to one another. As observed in experimental results our algorithm outperforms both the divide and conquer algorithm and the brute force algorithm. The algorithm is as follows:

1 Let $P = \{P_1, P_2, P_3, \ldots, P_n\}$ be a set of points.
$x_{max} = max(x_1, x_2, \ldots, x_n)$, $x_{min} = min(x_1, x_2, \ldots, x_n)$, $y_{max} = max(y_1, y_2, \ldots, y_n)$, $y_{max} = max(y_1, y_2, \ldots, y_n)$
$P_{max} = (x_{max}, y_{max})$, $P_{min} = (x_{min}, y_{min})$
2 $R = Dist(P_{min}, P_{max})$
Calculate $R_1 = 2 \times R$, $R_2 = 3 \times R$
$Sum =$ 1D Array
for $i = 1, \ldots, n$
$Sum[i] = R_1 \times Dist(P[i], P_i) + R_2 \times Dist(P[i], P_2)$
3 Sort the array Sum.
4 (*The closest pair of points will be at most 5 positions apart in Sum.*)
for $i = 1, 2, \ldots, n$
$P[i] =$ the point that corresponds to $sum[i]$
Find points which corresponds to $sum[i+1]$ to $sum[i+5]$
find the closest point to $P[i]$ among the above 5 points
return the Closest pair among all above pairs

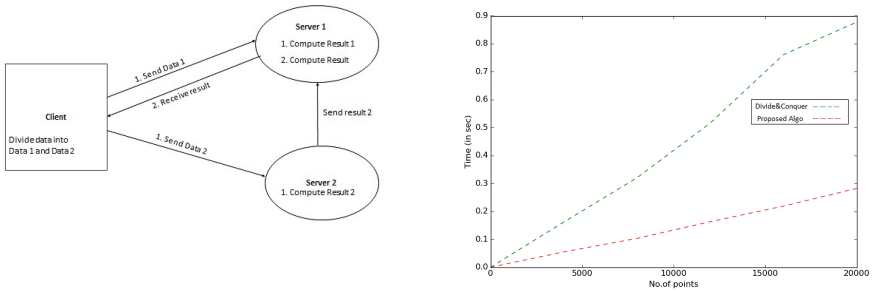
4.7 Experimental Results

In Fig. 2b it is observed that the time taken by the proposed algorithm is much less than the existing (and best known) divide and conquer version.



(a) Points (b) Encrypted Points

Fig. 1. The points are selected uniformly at random from the range 0–1000 units. Relative orientation and distances between the every pair of points is order preserved using our encryption scheme



(a) The Two-Server model of outsourcing

(b) Comparison between time taken by the divide and conquer and th proposed algorithm for computing closest pair of points

Fig. 2. Model for Two-Server cloud outsourcing and comparative analysis of the proposed scheme with the most efficient algorithm (Divide and conquer)

5 Conclusions and Discussion

Outsourcing of data and computations to the cloud servers with advanced computing resources is an effective application which is attracting numerous cloud users. However, when the server is untrusted, ensuring privacy of data (both input and output) and correctness of results becomes important. While outsourcing computations, it is naturally the case that they are computationally heavy and serve as a primitive step for more complex computations or data analytics. Computing closest pair of points is one such problem which involves repeated computations of Euclidean distances between different pair of points and is therefore a primitive step of many algorithms that involve grouping/clustering the data. A secure outsourcing algorithm for the problem is proposed with input and output privacy being provided by a novel order-preserving encryption algorithm

proposed in this paper. The result verification scheme has also been presented and the accompanying theoretical analysis proves its security. The client does only $O(n)$ work during this process. The adaptability of the proposed secure outsourcing scheme is shown with respect to the two-server cloud model. A novel algorithm for computing closest pair of points that outperforms its best known algorithm is also proposed. The verification algorithm needs to be improved to conform to the existing formal model for verifiable computing. This remains as the ongoing future work.

References

1. Benjamin, D., Atallah, M.: Private and cheating-free outsourcing of algebraic computations. In: 2008 Sixth Annual Conference on Privacy, Security and Trust (2008)
2. Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-preserving symmetric encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 224–241. Springer, Heidelberg (2009)
3. Tu, S., Kaashoek, M., Madden, S., Zeldovich, N.: Processing analytical queries over encrypted data. Proc. VLDB Endow. **6**, 289–300 (2013)
4. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS 2007) (2007)
5. Purushothama, B., Amberker, B.: Efficient query processing on outsourced encrypted data in cloud with privacy preservation. In: 2012 International Symposium on Cloud and Services Computing (2012)
6. Vyas, R., Singh, A., Singh, J., Soni, G., Purushothama, B.R.: Design of an efficient verification scheme for correctness of outsourced computations in cloud computing. In: Abawajy, J.H., Mukherjea, S., Thampi, S.M., Ruiz-Martínez, A. (eds.) SSCC 2015. CCIS, vol. 536, pp. 66–77. Springer, Heidelberg (2015). doi:10.1007/978-3-319-22915-7_7
7. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, p. 223. Springer, Heidelberg (1999)
8. Atallah, M.J., Pantazopoulos, K.N., Rice, J.R., Spafford, E.E.: Secure outsourcing of scientific computations. Adv. Comput. **54**, 215–272 (2002)
9. Seitkulov, Y.N.: New methods of secure outsourcing of scientific computations. J. Supercomputing **65**(1), 469–482 (2013)
10. Hohenberger, S., Lysyanskaya, A.: How to securely outsource cryptographic computations. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 264–282. Springer, Heidelberg (2005)
11. Tysowski, P.K.: Highly Scalable and Secure Mobile Applications in Cloud Computing Systems (Doctoral dissertation, University of Waterloo) (2013)
12. Hu, X., Tang, C.: Secure outsourced computation of the characteristic polynomial and eigenvalues of matrix. J. Cloud Comput. **4**(1), 1–6 (2015)
13. Mohassel, P.: Efficient and Secure Delegation of Linear Algebra. IACR Cryptology ePrint Archive, 605 (2011)
14. Wang, C., Ren, K., Wang, J., Urs, K.M.R.: Harnessing the cloud for securely solving large-scale systems of linear equations. In: 31st International Conference on Distributed Computing Systems (ICDCS), pp. 549–558. IEEE (2011)

15. Fiore, D., Gennaro, R.: Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 501–512. ACM (2012)
16. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 593–599. ACM (2005)
17. Bunn, P., Ostrovsky, R.: Secure two-party k-means clustering. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 486–497. ACM (2007)
18. Liu, D., Bertino, E., Yi, X.: Privacy of outsourced k-means clustering. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, pp. 123–134. ACM (2014)