Privacy-Preserving Data Mining in Homogeneous Collaborative Clustering

Mohamed Ouda, Sameh Salem, Ihab Ali, and El-Sayed Saad
Department of Communication Electronics and Computer Engineering, Helwan University, Egypt

Abstract: *Privacy concern has become an important issue in data mining. In this paper, a novel algorithm for privacy preserving in distributed environment using data clustering algorithm has been proposed. As demonstrated, the data is locally clustered and the encrypted aggregated information is transferred to the master site. This aggregated information consists of centroids of clusters along with their sizes. On the basis of this local information, global centroids are reconstructed then it is transferred to all sites for updating their local centroids. Additionally, the proposed algorithm is integrated with Elliptic Curve Cryptography (ECC) public key cryptosystem and Diffie-Hellman key exchange. The proposed distributed encrypted scheme can add an increase not more than 15% in performance time relative to distributed non encrypted scheme but give not less than 48% reduction in performance time relative to centralized scheme with the same size of dataset. Theoretical and experimental analysis illustrates that the proposed algorithm can effectively solve privacy preserving problem of clustering mining over distributed data and achieve the privacy-preserving aim.*

Keywords: *Privacy-preserving, secure multi-party computation, k-means clustering algorithm.*

Received December 20, 2013; accepted May 21, 2014; published online December 3, 2014

1. Introduction

Sharing of data between different parties can lead to mutual benefit, but due to privacy laws as in medical databases or privacy motivated by business interests, such sharing become difficult. The goal of privacy preserving of data mining is to get valuable information through mining process without disclosing the privacy of each party participating in mining scheme. Clustering analysis is one of the main tasks in data mining. Cluster analysis [12] is the process of dividing a set of data objects into groups or clusters that are meaningful, useful or both, such that objects within a given cluster are similar and dissimilar with other objects in different clusters. Clustering of data to groups is done according to some similarity measures of the data. Cluster analysis as one of data mining techniques is widely used in applications of financial affairs, marketing, insurance, medicine, chemistry, etc., clustering algorithms can be categorized into two classes, one class for algorithms based on point densities in which regions with a high density of points contribute to a cluster while the points in vicinity are arising, from the noise or distortion. DBScan is one of the most common of these algorithms [9].

The other class is based on the point distance approaches in which some initial points are added as the center of the clusters and then by defining a distance function like “Euclidean Distance” and changing the centers of the clusters they try to minimize the summation of the distance of the other points from the clusters. The final resulted points define the centers of the clusters. K-means clustering algorithm is one of the most outstanding methods of this class.

For distributed data mining, there are two forms of data: Horizontally partitioned data and vertically partitioned data. Horizontally partitioned data means that each site has complete information on a distinct set of entities and an integrated dataset consists of the union of these datasets. In contrast, vertically partitioned data has different types of information at each site; each has partial information on the same set of entities.

For horizontally partitioned database many distributed clustering algorithms are proposed recently such as K-Means [17] and DBDCL [15]. Miao and Genlin [25] proposed the distributed clustering algorithm, DK-Means, which improves K-DMeans algorithm. But the privacy concern in these clustering algorithms is not supported due to leakage of sensitive data. So, privacy preserving concern in distributed clustering is an important issue.

This paper develops a solution for privacy preserving K-means clustering for horizontally partitioned data using Elliptic Curve Cryptography (ECC) public key cryptosystem introduced in [27]. The idea of the global computation algorithm is based on DK-means distributed clustering [25] and applies Secure Multi-party Computation (SMC) protocol to protect real data of its own site from being transferring to other sites so that achieve privacy-preserving objective.

The paper is organized as follows: In section 2, we review in brief the related work in the area of privacy-preserving of K-means clustering algorithms. In section 3, privacy-preserving mechanisms used in proposed algorithm are presented. In section 4, we describe our proposed distributed clustering algorithm. In section 5, we present clustering validation measures to determine the optimum number of clusters for the dataset used. In section 6, the experimental evaluation of the proposed distributed clustering algorithm is presented. In section 7, we discuss the privacy preservation of the proposed algorithm. We conclude the paper in section 8 with a short summary and a few remarks.

2. Related Works

The need for secure information confidentiality during knowledge extraction process in distributed data environment is a very current area of research in scientific society. The privacy preserving clustering is classified into two main groups according to designing approach:

- One Group is the Randomization Method [1]: The concept of this group depends on the masking or perturbing the data set before applying the algorithm. Many algorithms are achieved by several researchers [19, 23, 26] using this approach.
- The Other Group is the Encryption Method: Encryption method mainly resolves the problems that people jointly perform mining tasks based on the private inputs they provide. This method is based on SMC protocol [21]. Several algorithms using this approach are introduced in [6, 8, 18]. The advantage of this method over the perturbing approach is that the data transformation is exact and secure.

According to how the data is organized, there are two different distributed privacy preserving data mining approaches, one on horizontally partitioned data and the other on vertically partitioned data.

Many research works studied the privacy-preserving of K-means clustering algorithms and considered various data partition models: Vertically partitioned data [29, 31], horizontally Partitioned data [16] and arbitrarily partitioned data [3, 24].

In SMC literature two basic adversarial models are defined: The first is semi-honest model, in which adversaries follow the protocol correctly, but can try to infer information of the other parties from the data they see during the execution of the protocol. The second is malicious model, in which malicious adversaries may do anything to infer secret information. They can abort the protocol at any time, send spurious messages, spoof messages, collude with other (malicious) parties, etc.

We will introduce in the paper the notion of privacy-preserving multi-party K-means clustering problem for horizontally partitioned data based on symmetric public key encryption scheme with semi-honest model as an adversarial model.

3. Privacy-Preserving Mechanisms

Now, a brief description of the tools used to achieve privacy preserving in the proposed algorithm will be presented.

- Diffe-Hellman Protocol [7]: Is the basic public key cryptosystem proposed for secret key sharing. In order to implement the Diffie-Hellman protocol we just need scalar multiplication P. In addition, if party A and party B first agree to use a specific curve, field size and type of mathematics, they then share the secret key by process as follows:

1. A and B each chose random private key k_a and k_b.
2. A and B each calculate (k_aP) and (k_bP) and send them to opposite side.
3. A and B both compute the shared secret key.

$$S_k = k_a(k_bP) = k_b(k_aP)$$

- ECC [2, 13]: Is an approach to public key cryptography in which every user has a public and a private key. Public key is used for encryption and private key is used for decryption. Only the particular user knows the private key whereas the public key is distributed to all users taking part in the communication. The main reason for choosing ECC is that it offers high level of security with small key size, e.g., elliptical curve encryption done using 160bit key gives the same level of security as given by RSA using 1024bit key. ECC is based on the algebraic structure of elliptic curves over finite fields [14]. Elliptic curves used in cryptography are typically defined over two types of finite fields: Prime fields F_p and binary extension fields F_{2m}. The ECC used in the proposed algorithm is defined over prime field F_p.

4. Proposed Privacy-Preserving Clustering on Distributed Databases

4.1. Proposed Model for Privacy Preserving

In our proposed algorithm we extend ECC and Diffie-Hellman key exchange to be multi parties cryptosystem for distributed environment dataset. In the proposed model as shown in Figure 1, master site and distributed slave sites on both ends of communication send a public key which can be seen by anyone. The public key is then combined with the private key to create a shared secret which, due to the underlying mathematics is the same on both sides. This shared secret is then used to hash a new key that can be used by either site (master and slave site) for encrypting and decrypting messages.

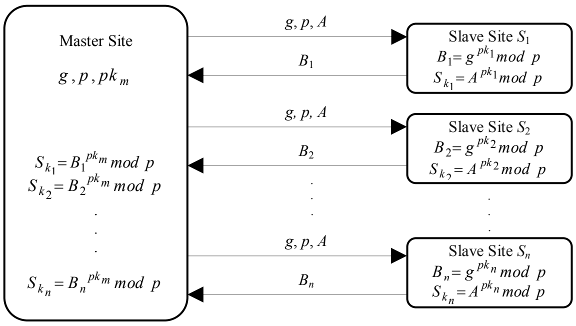

Figure 1. Shared secret key S_{ks} can be used by either master or slave sites for encrypting and decrypting messages.

Where p: Is a prime number on which the finite field F_p is defined, g: Is a base point taken from the elliptic group, pk_s, pk_m: Are private keys of slave site S_s and the master site S_m respectively selected from the interval [1, p-1], $1 \leq s \leq n$. B_s, A: Are public keys of slave site S_s and the master site S_m respectively. Then, the shared secret key S_{ks} is as follows:

$$S_{k_s} = mod(A^{pk_s}, p) = mod(mod(g^{pk_m}, p^{pk_s}), p) = mod(mod\ (g^{pk_m pk_s}, p)) = mod(mod(g^{pk_s}, p)^{pk_m}, p) = mod(B_s^{\ pk_m}, p)$$

Algorithm 1: Encryption algorithm.

1. *Let (G, E, D, M) be an ECC cryptography scheme, where G is an algorithm generating keys, E and D are the encryption and decryption algorithms and M is the message space.*
2. *Let s $1 \leq s \leq n$, (n= Number of sites/ parties) and k is the number of clusters in each site.*
3. *Each site s has k clusters; each cluster has a size m_i^s, $1 \leq i \leq k$ and centroid C_i^s of dimension space r where $C_i^s = \{c_{i1}^s, c_{i2}^s, ..., c_{ir}^s\}$, (i is the cluster number at site s).*
4. *At the master site and each slave site the public and private key pair of ECC algorithm is generated. Key pair for a slave site s is: (B_s, pk_s) $1 \leq s \leq n$ and for the master site is: (A, pk_m).*
5. *Each slave party/site s exchanges public encryption key with the master site.*
6. *The shared secret key S_k is generated for each slave site s, $1 \leq s \leq n$ and the master site. This shared secret key is then used by both master and slave site for encrypting and decrypting of messages. Each slave site s calculate its shared secret key as follows:*

   $$S_{k_s} = mod(A^{pk_s}, p) = mod(g^{pk_m pk_s}, p)$$

   Where A is the public key of the master site:

   $$A = mod(g^{pk_m}, p)$$

   While the shared secret key at the master site corresponding each slave site is calculated as follows:

   $$S_{k_s} = mod(B_s^{\ pk_m}, p),\ B_s = mod(g^{pk_s}, p)$$

   Where B_s is the public key of sites.

   $$S_{k_s} = mod((g^{pk_s})^{pk_m}, p) = mod(g^{pk_m pk_s}, p)$$

7. *Given a clusters size array $m^s \in M$ and a cluster centroid $C_i^s \in M$, of cluster i at site s as plaintext messages where:*

   $$C_i^s = \{c_{i1}^s, c_{i2}^s, ..., c_{ir}^s\},\ 1 \leq i \leq k$$
   $$m^s = \{m_1^s, m_2^s, ..., m_k^s\}$$

8. *The encrypted values are computed as:*

   $$E_{S_{k_s}}(C_i^s) = mod(C_i^s g^{pk_m pk_s}, p),\ 1 \leq i \leq k \ \ and\ 1 \leq s \leq n \tag{1}$$

   $$E_{S_{k_s}}(m^s) = mod(m^s g^{pk_m pk_s}, p),\ 1 \leq s \leq n \tag{2}$$

Algorithm 2: Decryption Algorithm.

To decrypt:

$$E_{S_{k_s}}(C_i^s) = mod(C_i^s g^{pk_m pk_s}, p)\ \ and\ \ E_{S_{k_s}}(m^s) = mod(m^s g^{pk_m pk_s}, p)$$

at master site the decryption key is calculated which represents the inverse value of the shared key S_k, then,

$$D_{S_{k_s}} = mod((g^{pk_m pk_s})^{-1}, p) = (S_{k_s})^{-1}$$

such that:

$$(S_{k_s}^{\ -1}(E_{S_{k_s}}(C_i^s))) = (mod((g^{pk_m pk_s})^{-1}, p)(mod(C_i^s g^{pk_m pk_s}, p)) = mod((g^{pk_m pk_s})^{-1} g^{pk_m pk_s} C_i^s, p) = mod(C_i^s, p)$$

Then,

$$D_{S_{k_s}}(E_{S_{k_s}}(C_i^s)) = mod(C_i^s, p) = C_i^s,\ 1 \leq i \leq k\ \ and\ 1 \leq s \leq n \tag{3}$$

Then, also:

$$D_{S_{k_s}}(E_{S_{k_s}}(m^s)) = m^s,\ 1 \leq s \leq n \tag{4}$$

As $|F_p|$ is the order of the finite group F_p then, $x^{|Fp|}=1$ for all x in F_p, as established from Lagrange's theorem in group theory [10]. The order$|F_p|$, of the group F_p is known for all sites.

- The value $D_{S_{k_s}} = mod((g^{pk_m pk_s})^{-1}, p)$ at the master client will be calculated as follows:

  It is known its private key pk_m and the public key B_s of slave site s, $B_s = mod(g^{pk_s}, p)$ then:

  $$mod((g^{pk_s}, p)^{|Fp|-pk_m}, p) = mod(g^{|Fp|pk_s - pk_m pk_s}, p) = mod(g^{|Fp|pk_s} g^{-pk_m pk_s}, p) = mod(1^{pk_s} g^{-pk_m pk_s}, p) = mod(g^{-pk_m pk_s}, p) = mod((g^{pk_m pk_s})^{-1}, p)$$

- The value $D_{S_{k_s}} = mod((g^{pk_m pk_s})^{-1}, p)$ at the slave site will be computed as follows:

  It is known its private key pk_s and the public key of master site $A = mod(g^{pk_m}, p)$ then:

  $$mod((g^{pk_m}, p)^{|Fp|-pk_s}, p) = mod(g^{pk_m |Fp| - pk_m pk_s}, p) = mod(g^{|Fp|pk_m} g^{-pk_m pk_s}, p) = mod(1^{pk_m} g^{-pk_m pk_s}, p) = mod((g^{pk_m pk_s})^{-1}, p)$$

The result of decryption from one site s is k centroids of k clusters and their corresponding array sizes m^s to produce the global centroids components as per Equation 6 at the master site. Each site learns nothing about other sites. Since, the K-means algorithm is performed locally for every site/party, this enables solutions where the communication cost is independent of the size of the database.

4.2. Proposed Algorithm for Distributed Clustering

The proposed algorithm in this paper consists of two levels local and global and four steps which are demonstrated in Figure 2 are as follows:

1. Local clustering.
2. Extraction of local properties (local centroids and corresponding local clusters size).
3. Determining global cluster centroids based on the local received values.
4. Updating of all local cluster models.

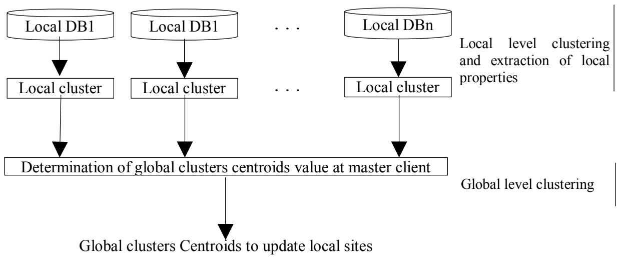
Local DB1
Local DB1
. . .
Local DBn
Local level clustering and extraction of local properties
Local cluster
Local cluster
. . .
Local cluster
Determination of global clusters centroids value at master client
Global level clustering
Global clusters Centroids to update local sites

Figure 2. Steps of clustering at distributed databases.

The mathematical model of distributed data sets over horizontal partition is as follows:

- Suppose $DB_i(1\leq i\leq n)$ among the data sets DB_1, DB_2, …, DB_n located at different sites P_1, P_2, ..., P_n (i.e., n-divisions) as the partial database and $DB = DB_1 \cup DB_2 \cup ... \cup DB_n$ as the overall situation database. Each database DB_i has r attributes and different number of entities. A pre-processing work is done for normalization of all values of distributed dataset at each site before implementation the proposed algorithm.
- We also assume that the adversary model is semi-honest in which parties follow the execution requirement of the protocol but may use what they see during the execution to compute more than they need to know.
- The distance function used in this work is the standard Euclidean distance which is defined as:

$$D(X\ ,Y) = \sqrt{\sum_{i=1}^{r}(X\ [i\ ]\text{-}Y\ [i\ ])^2} \tag{5}$$

Where r is the dimension space of an instance X, $X[i]$ denote the i^{th} component value of data object X and $D(X, Y)$ is the distance between two data objects X, Y.

The process of proposed algorithm is shown below in Figure 3.

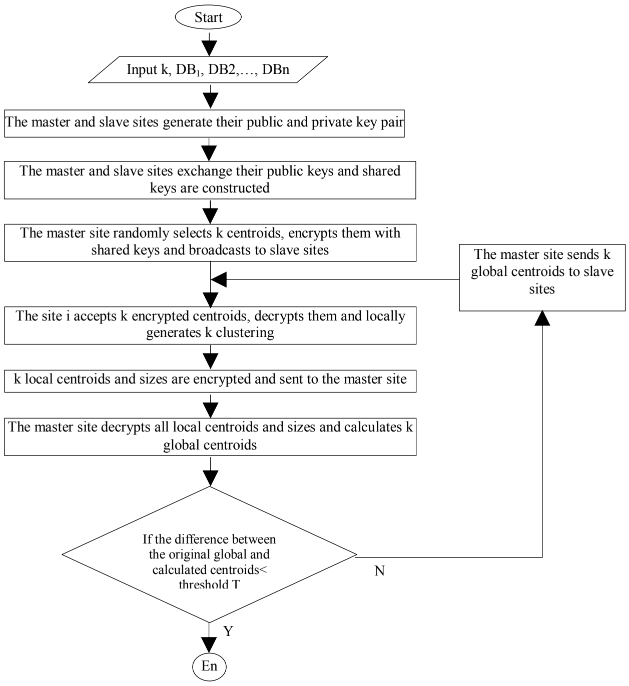
Start
Input k, DB1, DB2,…, DBn
The master and slave sites generate their public and private key pair
The master and slave sites exchange their public keys and shared keys are constructed
The master site randomly selects k centroids, encrypts them with shared keys and broadcasts to slave sites
The master site sends k global centroids to slave sites
The site i accepts k encrypted centroids, decrypts them and locally generates k clustering
k local centroids and sizes are encrypted and sent to the master site
The master site decrypts all local centroids and sizes and calculates k global centroids
If the difference between the original global and calculated centroids< threshold T
N
Y
En

Figure 3. The process of proposed algorithm.

Algorthim 3: The integrated privacy preserving algorithm of distributed K- Means clustering.

Input: *DB_1, DB_2, ..., DB_n at n sites, each of d_j data objects, $j\in\{1, ..., n\}$, each data object $X=x_1$, x_2, ..., x_r of r space dimension*, $r>1$, *k clustering value.*

Output: *k global centers, each of which is the centriod of objects belonging to the same cluster.*

1. *The first site is considered as the master site and in which an initial k centers are randomly chosen as the a global centroids of the k clusters which are broadcast to all local sites. The initial k centers are $G=\{c_1, c_2, …,c_k\}$.*
   In addition (e, d) *key pair as the encryption and decryption keys of ECC algorithm is also generated at master and slave sites. Master and slave sites exchange their public encryption key to produce shared secret S_{k_j} which is used for encryption and decryption between the master and slave site j.*
2. *In each local site, every data object, finding its closest center c_i and then is assigned to coressponding cluster i, $j\in\{1, ..., k\}$.*
3. *The new set of centers at site j. $C_j'= \{c_1', c_2', …, c_k'\}$ and corresponding set of clusters' sizes $M_j'= \{m_1', m_2', …, m_k'\}$ are calculated based on the cluster assignment in step* 2.
4. *All local centroids C_j' and corresponding sizes M'_j are sent from local sites to the master site after encryption every set as per Equations* 1, 2.
5. *At master site a decryption process is done for all received C_j' and M_j', as per Equations* 3, 4 *then a new set of global centroids G' are calculated as per Equation 6.*
6. *If the difference between the G' and G is less than a preset threshold T.*
7. *Then terminate and output G and M_j' for every site.*
8. *If not go to step* 2 *and replace C_j' by G'.*

Generally speaking, we randomly choose the initial position of these k centers $G=\{c_1, c_2, ..., c_k\}$ in step 1. In steps 2 to 7 we first assign each data object X to the cluster whose center is close to it, where we use Euclidean distance as a metric distance as per Equation 5. Then, in step 3 recalculate a new set of k clusters centroids $C_j'= \{c_1', c_2', …, c_k'\}$ at each site j based on the cluster assignment in step 2 and calculate corresponding set of cluster's size $M_j'= \{m_1', m_2', …, m_k'\}$. The new centroid of each cluster is calculated by using the arithmetic mean method, i.e., by computing the arithmetic mean of the data objects in the cluster. Let the number of data instances belonging to a cluster be D. For a property x of the data instances, sum $\overrightarrow{S_i} = \sum_{g=1}^{D} \overrightarrow{x_g}$, mean $\overrightarrow{M_i} = \frac{S_i}{D}$, $1 \leq i \leq k$.

In step 4 the new centroids and corresponding sizes are sent from all local sites to the master site. In step 5 a new set G' of global centers are calculated. In step 6 we decide if the new set of global centers is good enough by checking in the difference between the new set of centers G' and the old set of centers G. If it is small enough within a given threshold, we terminate the algorithm and return G as the final result as in step 7. Otherwise, in step 8, we will use the new set of centers G' and iterate the process. The computation of distance between two sets of centers is done using, Euclidean distance function as per Equation 5. In step 5 according to the received information, the master site calculates the overall clustering centroid components (global components) C_{ij}, $j\in\{1, 2, ..., C_k\}$ of global array C_g of dimension r, $C_g\in\{C_1, C_2, ..., C_k\}$ as follows [27]:

$$C_{ij}=\frac{c_{ij}^{1}\times m_{i}^{1}+c_{ij}^{2}\times m_{i}^{2}+...+c_{ij}^{s}\times m_{i}^{s}}{m_{i}^{1}+m_{i}^{2}+...+m_{i}^{s}} \tag{6}$$

Where c_{ij}^{1} is the component j of the centroid of cluster i at site 1 and m_i^1 is the size of cluster i at site 1. The global element C_{ij} of global centroid C_g means that every j component of local centroid for cluster i is multiplied by its size and then the sum of the weighed elements is divided by the sum of cluster size i at all sites. To illustrate the above formula, let us have the following example: 2 sites, each site has 2 clusters and each cluster centroid has two attributes. Then, for the first site cluster centroids and corresponding sizes are $\{c_{11}= (2, 3), m_{11}= 4\}$, $\{c_{12}= (1, 6), m_{12}= 5\}$ and for the second site are $\{c_{21}= (1, 2), m_{21}= 2\}$, $\{c_{22}= (2, 5), m_{22}= 3\}$.

The first global centroid elements of first global cluster C_{g1} are calculated as follows:

$$c_{11}=\frac{2\times 4+1\times 2}{4+2}=\frac{10}{6},\quad c_{12}=\frac{3\times 4+2\times 2}{4+2}=\frac{16}{6}$$

Then, $C_{g_1}=(\frac{10}{6},\frac{16}{6})$ and for the second global centroid elements of second global cluster C_{g2}:

$$c_{21}=\frac{1\times 5+2\times 3}{5+3}=\frac{11}{8},\quad c_{22}=\frac{6\times 5+5\times 3}{5+3}=\frac{45}{8}$$

Then, $C_{g_2}=(\frac{11}{8},\frac{45}{8})$

To check the optimum number of clustering for the data set used in distributed environment, four validation indices are applied.

5. Clustering Validation Measures

The optimum clustering number is the goal of clustering to make objects within the same cluster similar and objects in different clusters distinct, internal validation measures are used for that, which are based on the following two criteria [28, 32]:

1. Compactness: It measures how closely related the objects in a cluster are.
2. Separation: It measures how distinct or well-separated a cluster is from other clusters.

The most common used indices are: Index (I) [22], Calinski-Harabasz index (CH) [4], Davies-Bouldinindex (DB) [5] and The Xie-Beniindex (XB) [30]. The optimum clustering number is represented as the maximum or minimum value as per definition formula of the corresponding method.

- *I* index:

$$I=(\frac{1}{NC}.\frac{\sum_{x\in D}d(x,c)}{\sum_{i}\sum_{x\in C_i}d(x,c_i)}.\max_{i,j}d(c_i,c_j))^p$$

- *CH* index:

$$CH=\frac{\sum_{i}n_i d^2(c_i,c)/(NC-1)}{\sum_{i}\sum_{X\in C_i}d^2(x,c_i)/(n-NC)}$$

- *DB* index:

$$DB=\frac{1}{NC}\sum_{i}\max_{j,j\neq i}\{[\frac{1}{n_i}\sum_{x\in C_i}d(x,c_i)+\frac{1}{n_j}\sum_{x\in C_j}d(x,c_j)]/d(c_i,c_j)\}$$

- *XB*-index:

$$XB=[\sum_{i}\sum_{x\in C_i}d^2(x,c_i)]/[n.\min_{i,j,i\neq j}d^2(c_i,c_j)]$$

Where *D*: Dataset, *n*: Number of objects in *D*. *c*: Center of *D*. *p*: Number of attributes in *D*. *NC*: Number of clusters. C_i: The i^{th} cluster, n_i: Number of objects in C_i. c_i: Center of C_i. $d(x, y)$: Distance between x and y.

Validation Index Measure	Optimal Value
I	Max
CH	Max
DB	Min
XB	Min

6. Experimental Evaluation

We evaluated our proposed approach based on different real-world datasets as per Table 1 [20]. The data objects sets were generated on each local site independently. For the central reference clustering we used the union of the local object sets. As we suppose that this central clustering is optimal, we measure the performance time of our proposed approach w.r.t. the central encrypted clustering. We varied both the number of data objects and the number of client sites. We compared proposed algorithm to a single run of *k* means clustering on all data objects. In order to evaluate the proposed algorithm, we carried out the local clustering sequentially. We collected all encrypted representatives of all local runs, and then applied a global clustering on these representatives after decryption process. For all these steps we always used the same computer. The overall runtime was formed by adding the time needed for the global clustering to the maximum time needed for the local clustering. All experiments were developed using C# standard Edition 2010 on Intel® Core2 Duo, 2.0 GHz, 4 GB RAM machine.

Table 1. Data sets.

Data Set Name	Attribute Characteristics	Number of Instances	Number of Attributes	Area
Adult	Categorical, Integer	6000	13	Social
Breast Cancer	Real	600	10	Life

Before starting the proposed algorithm the optimum number of clusters *K* is searched with the aid of 4 clustering validation measures, which are applied on datasets used, Adult and Breast Cancer respectively. These measure metrics are:

1. I [22].
2. CH [4].
3. DB [5].

4. XB [30].

As shown in Tables 2, 3 the optimum number of clusters is 2 for the used datasets, Adult and Breast Cancer respectively. Our proposed algorithm is performed with cluster number *k* equal 2. We did distributed clustering with and without proposed encryption to know how the proposed encryption algorithm can affect the system performance.

Table 2. Clustering validation of 4 measure metrics for adult dataset (normalized values).

K	I	CH	DB	XB
	Optimal Value(Max)	**Optimal Value(Max)**	**Optimal Value(Min)**	**Optimal Value(Min)**
2	**1**	**1**	**0**	**0.43**
3	0.6	0.6	1	0.46
4	0.58	0.42	0.66	0.54
5	0.48	0.41	0.38	0.59
6	0.44	0.29	0.24	0.61
7	0.36	0.3	0.26	0.68
8	0.56	0.16	0.14	1

Table 3. Clustering validation of 4 measure metrics for breast cancer dataset (normalized values).

K	I	CH	DB	XB
	Optimal Value(Max)	**Optimal Value(Max)**	**Optimal Value(Min)**	**Optimal Value(Min)**
2	**1**	**1**	**0.01**	**0.21**
3	0.33	0.65	1	0.27
4	0.62	0.53	0.42	0.41
5	0.84	0.59	0.33	0.46
6	0.77	0.71	0.32	0.68
7	0.24	0.5	0.2	0.77
8	0.21	0.47	0.39	1

In Tables 4, 5 the execution time is measured for distributed dataset with and without proposed encryption scheme and the performance time for centralized encrypted dataset is compared with the proposed distributed one.

Table 4. Execution time for distributed/ centralized breast cancer dataset.

No. of Sites	Dataset Size in Thousands of Bytes	Execution Time of Dataset (ms)		
		Execution Time of Distributed Dataset		**Execution Time of Centralized Encrypted Dataset**
		Without Proposed Encryption	**With Proposed Encryption**	
2	19,200	502	577	1242
3	28,800	511	584	1423
4	38,400	541	600	1613
5	48,000	599	647	1790
6	57,600	612	676	2007

Table 5. Execution time for distributed/centralized adult dataset.

No. of Sites	Dataset Size in Thousands of Bytes	Execution time of dataset (ms)		
		Execution Time of Distributed Dataset		**Execution Time of Centralized Encrypted Dataset**
		Without Proposed Encryption	**With Proposed Encryption**	
2	192,000	2552	2642	5144
3	288,000	2632	2651	7548
4	384,000	2679	2775	9677
5	480,000	2708	2802	11852
6	576,000	2736	2950	14090

Figures 4, 5 show the execution time of encrypted centralized system compared to the proposed encrypted distributed one. Comparing the execution time for encrypted distributed datasets of 6 sites with centralized one, the reduction of execution time in the proposed distributed system of 6 sites of total size 600 record of breast cancer dataset is not less than 53% the centralized one with the same dataset size. But, for the case of adult dataset proposed distributed system of 6 sites of total size 6000 record the reduction of execution time is more than 48% of centralized one. This is shows how encrypted centralized clustering can affect the performance of the system comparing it with the proposed distributed encrypted scheme.

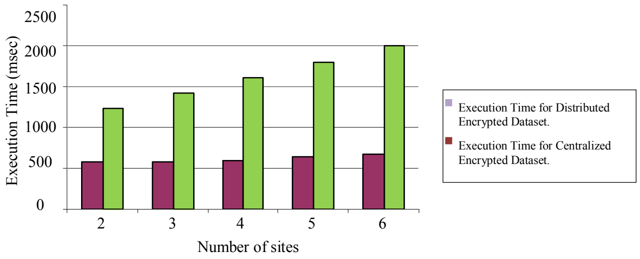

Figure 4. Comparison of execution time for proposed encrypted distributed Breast Cancer datasets and centralized encrypted one.

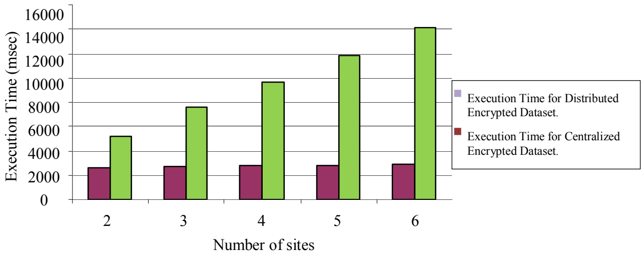

Figure 5. Comparison of execution time for proposed encrypted distributed Adult datasets and centralized encrypted one.

As it is shown in Figures 6, 7 the difference in execution time due to the proposed encryption scheme on distributed databases does not exceed than 15% of execution time without encryption for Breast Cancer dataset in 6 sites with 2 cluster of total size 100 record at each site, but for adult dataset the difference in execution time does not exceed 8% for dataset of size 1000 record at each site, which means that the effect of proposed encryption scheme comparable to execution time of the system without encryption is acceptable.

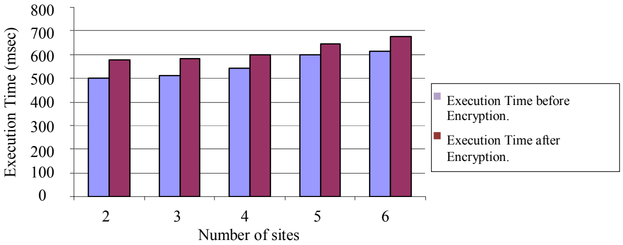

Figure 6. Effect of proposed encryption scheme on execution time of distributed breast cancer datasets.

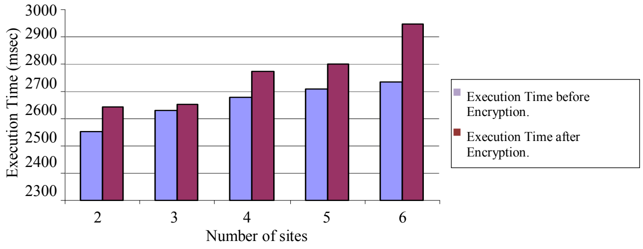

Figure 7. Effect of proposed encryption scheme on execution time of distributed adult datasets.

7. Discussion

- Privacy Preserving Analysis: As the data objects at each local site are normalized and then clustered locally using K-means clustering algorithm, so no interactions between parties. Thus, there is no question of privacy being revealed or compromised. We reduce the problem to that of privately computing smaller sub problems and show how to compose them together in order to obtain a complete solution of privacy preserving K-means clustering. This composition is shown to be secure in [11].

  a. Each Party P_s encrypts its local clusters size $M'_s=\{m^s_1,m^s_2,...,m^s_k\}$ and corresponding clusters centroids $C'_s=\{c^s_1,c^s_2,...,c^s_k\}$ as per Equations 1, 2. These encrypted values $E_{k_s}(C^s_i)$ and $E_{k_s}(m^s)$ are transmitted to the master site for global clustering. So, the output of each site is securely transmitted to the master site to compute the global clustering centroids without leaking any information about the private data of a party except its output.

  b. ECC is semantically secured due to the difficulty of the elliptic curve discrete logarithm problem. Also using ECC in combination with Diffie-Hellman protocol is believed to make public key encryption more secure.

  Master site, which decrypts, the cluster centroid $E_{k_s}(C^s_i)$ and its size $E_{k_s}(m^s_i)$ as well as the slave sites which decrypt the global centroids produce accurate results with ECC cryptosystem.

- The Complexity Analysis of the Protocol:

  1. The communication cost.
  2. Let us use α to denote the number of bits of each cipher text and "*n*" is the total number of parties/ sites. "*k*" is the number of clusters (centroids), and "*r*" is the space dimension of each centroid cluster. The total communication cost is $n\alpha(1+k)$ from step 4 in the proposed algorithm.
  3. The computational cost is affected by:

     a. The generation of *n* cryptographic key pair.
     b. The total number of $n(1+k)$ encryptions and $n\alpha(1+k)$ decryptions.
     c. Complexity for local *k* means algorithm is $O(lrdk)$, where *l* is the number of iterations, *r* is the number features/attributes, *d* is the number of data objects in a local site and *k* is the number of clusters.
     d. Additional computations as $nk(r+1)$ additions, *nkr* multiplication and *rk* division.

Hence, the complexity of *n* parties is dominant for not only the other computational costs but also for communication costs too. Consequently, the overall Complexity of the proposed model=$O(lrdkn)$.

Therefore, the proposed algorithm shows rising in efficiency due to decrease in time complexity.

The proposed algorithm reduces the time complexity mainly in two aspects.

- *First*: Global centroids $G_g=\{c_{g1},c_{g2},...,c_{gk}\}$ are quickly generated, since the K-means algorithm executed locally for every party P_s, this enables solutions where the communication cost is independent of the size of the database and greatly cut down communication costs comparing with centralized data mining which needs to transfer all data into central data warehouse to perform data mining algorithm, as shown in Figure 5.
- *Second*: The length of encryption-decryption key size is shorter than other public key encryption methods (e.g., RSA) with the same level of security.

8. Conclusions

In this paper, the need of privacy preserving in distributed environment has been motivated. The data are locally clustered at each site and only encrypted aggregated information about the local data is transmitted to the master site. This aggregated information consists of a set of local clusters centroids and corresponding sizes. On the basis of this local information, global clustering centroids are reconstructed. The created global centroids are sent to all clients, who use this information to update their own local clusters. This solution depends mainly on integration of ECC public key cryptosystem and Diffie-Hellman key exchange which is semantically secured. The proposed distributed encrypted scheme can add overhead increase not more than 15% in performance time relative to non encrypted distributed scheme but give not less than 48% reduction in performance time relative to centralized scheme with the same size of dataset. Experimental results show that proposed algorithm has good capability of privacy preserving, accuracy, efficiency and relatively comparable to centralized approach.

References

[1] Agrawal R. and Srikant R., "Privacy-Preserving Data Mining," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 439-450, 2000.

[2] Amara M. and Siad A., "Elliptic Curve Cryptography and its Applications," *in Proceedings of the 7th International Workshop on System*, *Signal Processing and their Application*, Tipaza, Algeria, pp. 247-250, 2011.

[3] Bunn P. and Ostrovsky R., "Secure Two-Party k-Means Clustering," *in Proceedings of the 14th ACM Conference on Computer and Communications Security*, Virginia, USA, pp. 486-497, 2007.

[4] Calinski T. and Harabasz J., "A Dendrite Method for Cluster Analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1-27, 1974.

[5] Davies D. and Bouldin D., "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224-227, 1979.

[6] Dhillon I. and Modha D., "A Data Clustering Algorithm on Distributed Memory Multiprocessors," *Large-Scale Parallel Data Mining*, 2000.

[7] Diffie W. and Hellman M., "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[8] Duand W. and Atallah M., "Privacy-Preserving Cooperative Statistical Analysis," *in Annual Computer Security Applications Conference ACSAC*, Louisiana, USA, pp. 102-110, 2001.

[9] Forman G. and Zhag B., "Distributed Data Clustering can be Efficient and Exact," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 34-38., 2000.

[10] Gallian J., *Contemporary Abstract Algebra*, Boston: Houghton Mifflin, 2006.

[11] Golwasser S. and Micali S., "Probabilistic Encryption," *the Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, 1984.

[12] Han J. and Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

[13] Hankerson D., Menezes A., and Vanstone S., *Guide to Elliptic Curve Cryptography*, Springer-Verlag, 2004.

[14] Hasegawa T., Nakajima J., and Matsui M., "A Practical Implementation of Elliptic Curve Cryptosystems Over GF(P) on A 16-Bit Microcomputer," *in Proceedings of the 1st International Workshop on Practice and Theory in Public Key Cryptography*, Pacifico Yokohama, Japan, pp. 182-194, 1998.

[15] Januzaj E., Kriegel P., and Pfeifle M., "DBDC: Density Based Distributed Clustering[c]," *in Proceedings of the 9th International Conference on Extending Database Technology*, Crete, Greece, pp. 88-105, 2004.

[16] Jha S., Kruger L., and McDaniel P., "Privacy Preserving Clustering," available at: http://siis.cse.psu.edu/pubs/esorics05.pdf, last visited 2013.

[17] Kantabutra S. and Couch L., "Parallel K-Means Clustering Algorithm on Nows," *National Electronics and Computer Technology Center Technical Journal*, vol. 1, no. 6, pp. 243-247, 2000.

[18] Kargupta H., Huang W., Sivakumar K. and Johnson E., "Distributed Clustering using Collective Principal Component Analysis," *Knowledge and Information Systems*, vol. 3, no. 4, pp. 405-421, 2001.

[19] Klusch M., Lodi S., and Moro G., "Distributed Clustering based on Sampling Local Density Estimates," available at: http://www-ags.dfki.uni-sb.de/~klusch/papers/ijcai03-KDEC-paper.pdf, last visited 2003.

[20] Kohavi R. and Becker B., "UCI Repository of Machine Learning Databases," available at: http://archive.ics.uci.edu/ml/datasets.html, last visited 2013.

[21] Linedell Y. and Pinkas B., "Secure Multiparty Computation for Privacy-Preserving Data Mining," *the Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59-98, 2009.

[22] Maulik U. and Bandyopadhyay S., "Performance Evaluation of Some Clustering Algorithms and Validity Indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650-1654, 2002.

[23] Merugu S. and Ghosh J., "Privacy-Preserving Distributed Clustering using Generative Models," *in Proceedings of the 3rd International Conference on Data Mining*, Florida, USA, pp. 211-218, 2003.

[24] Meskine F. and Bahloul S., "Privacy Preserving K-means Clustering: A Survey Research," *the International Arab Journal of Information Technology*, vol. 9, no. 2, pp. 194-200, 2012.

[25] Miao Z. and Genlin J., "DK-Means-An Improvement of Distributed Clustering Algorithm K-Dmeans," available at: http://d.g.wanfangdata.com.cn/Periodical_jsjyjyfz2007z2017.aspx, last visited 2013.

[26] Oliveira S. and Zaiane O., "Privacy Preserving Clustering by Data Transformation," *Journal of Information and Data Management*, vol. 1, no. 1, pp. 304-318, 2010.

[27] Pardo J., "An Introduction to Elliptic Curve Cryptogaphy," *Introduction to Cryptography with Maple*, 2013.

[28] Tan P., Steinbach M., and Kumar V., *Introduction to Data Mining*, USA: Addison-Wesley Longman, Inc, 2005.

[29] Vaidya J. and Clifton C., "Privacy-Preserving k-Means Clustering over Vertically Partitioned Data," *in Proceedings of the 9th ACM SIGDD International Conference on Knowledge Discovery and Data Mining*, Illinois, USA, pp. 206-215, 2003.

[30] Xie X. and Beni G., "A Validity Measure for Fuzzy Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841-847, 1991.

[31] Yi X. and Zhang Y., "Equally Contributory Privacy-Preserving K-Means Clustering Over Vertically Partitioned Data," *Information Systems*, vol. 38, no. 1, pp. 97-107, 2013.

[32] Zhao Y. and Karypis G., "Evaluation of Hierarchical Clustering Algorithms for

Document Datasets," *in Proceedings of the 11th International Conference on Information and Knowledge Management*, Virginia, USA, pp. 515-524, 2002.

Mohamed Ouda PhD student in Communications and Computer Engineering Department, Helwan University, Egypt. His research interests include machine learning, data mining, and database security.

Sameh Salem graduated with a BSc and MSc degrees in communications and electronics engineering, both from Helwan University, Egypt, in 1998 and 2003, respectively. In 2008, He received the degree of PhD in engineering from Department of Electrical Engineering and Electronics, The University of Liverpool, UK. His research interests include clustering algorithms, machine learning, data mining, parallel computing, and cloud computing. In 2008, He was appointed as assistant professor in Department of Electronics, Communication and Computer Engineering, Faculty of Engineering, Helwan University, Egypt. Also, He is selected to be coordinator and academic advisor at Department of Communication and Information Technology, Uninettuno University (Italy) in corporation with Faculty of Engineering, Helwan University (Egypt). Furthermore, He is reviewing several proposals and research projects at the National Telecommunication Regulatory Authority (NTRA)-Egypt. In 2014, He is promoted to be Associate Professor. Currently, he is Honorary Research Fellow at the Department of Electrical Engineering and Electronics, The University of Liverpool, UK.

Ihab Ali obtained his BSc, MSc and PhD degrees at 1985, 1991 and 1997 respectively, all in communications Engineering from Helwan University, Egypt. He is a senior member of IEEE. He is currently the head of Communications Engineering Department, Helwan University, Egypt.

EL-Sayed Saad is Professor of Electronic Circuits, Faculty of Engineering, Helwan University, Egypt. International scientific member of the ECCTD. Member of the national radio science committee. Member of the European Circuit Society (ECS). Inventor of Scaad's single amplifier SC structure. Engineering Consultant for the Supreme Council of Universities.