Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLICATIONS OF PRIVACY PRESERVING K-MEANS CLUSTERING OVER OUTSOURCED DATA ON CLOUD PLATFORM

[1]ANURAG, [2]DEEPAK ARORA, [3]UPENDRA KUMAR

[1]Amity Institute of Information Technology, Amity University, Uttar Pradesh, Lucknow Campus, India

[2]Department of Computer Science & Engineering, Amity University Uttar Pradesh, Lucknow , India

[3]Department of Computer Science, Birla Institute of Technology, Patna, India

[1]anurag.smit@gmail.com, [2]deepakarorainbox@gmail.com, [3]upendrakumarphdp@gmail.com

ABSTRACT

Data Mining has gained attention nowadays in the field of sales, marketing, insurance and healthcare applications to name a few. Organizations aspire to perform mining operations on their joint datasets for gaining trade benefits while hiding own sensitive information. Owing to huge resource consumption and less computational power, they often prefer to outsource their data on the cloud platform for entire computation. As there is a risk of exposing the organization's sensitive data from various mistrusted parties involves in it, privacy becomes one of the major challenging issues in cloud computing. Authors have proposed an algorithm were cloud server applies k means clustering on encrypted data sets. A Trusted Party is assumed for key distribution and management. Computations between each party are either performed mutually or via Trusted Authority which involving exchange of sensitive data transfer of each participating parties. Complexity of the algorithm has been analyzed and compared with the existing approach and found that it is linearly depends upon various parameters settings and hence is a better approach while maintaining authenticity and data confidentiality between various participating parties during the mining process.

Keywords: *Privacy Preserving Data Mining, Pailler Homomorphic Encryption, K-Means Clustering, Cloud Platrorm, Use Case Diagram.*

1. INTRODUCTION

Cloud Computing allows the user in accessing computing resources and services on demand without having to buy its own infrastructure [1]. Different organizations often enhance their business activities on cloud. It has various advantages over traditional computation such as improved productivity, reduction in infrastructure and maintenance costs [2], powerful distributed capacity and capability to handles large amount of data [3]. It improves business strategies objectives. It is most prominent to increase return on the capital thus provide customer satisfaction, improve quality and efficiency, create a high performance culture and optimize customer profitability [4]. So, organizations outsource data to the cloud server for performing huge and efficient computation. As the trends in IT industry to outsource data to the cloud platform has been growing tremendously, risk of unveiling organizations sensitive data to unauthorized parties has also been increasing. Security and confidentiality has been one of the main issues in data mining as massive data in the cloud platform are vulnerable to be retrieve and misuse by the various mistrusted parties. So, secure computation on the cloud assisted platform has become the one of the most challenging issues nowadays.

Privacy Preserving Data Mining deals with confidentiality of sensitive data on the cloud server during mining process. In recent years, there has been a growing trend in distributed data mining applications in which datasets are physically distributed among multiple sites often across different geographical locations. These sites collectively collaborate together during mining operation on their joint datasets. There are two types of Distributed database-horizontally partitioned data and vertically partitioned data. In horizontal database, each sites contain the same sets of attributes in different transactions, while is vertically partitioned data, each dataset contains different attributes in same set of transactions. Each

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parties wishes to apply data mining for improving trade benefits without revealing their sensitive data to other parties. This scenario is analogous to Yao millionaire problem [5] in which each party wants to know who more millionaires are without revealing their net income to each other. Now let us consider a situation in which each organization maintains the record containing the list of different items bought by the different age groups, occupations, gender and region of customers per day on a difference season during each transaction. They wishes to apply data mining operations on the joint datasets for further improving decision making process for increasing sales of the particular items according to certain season and age groups of customer and hence in turn increases profit, but does not wants to reveal its sensitive information due to danger in misuse of confidential data either to the malicious cloud server or to the various other parties associated with it. Two types of models are deployed - Semi-honest and malicious. Semi-honest model follows the protocol honestly, but are curious in learning the sensitive data of the data owner. Malicious model does not honestly follow the protocol specification. In addition to the above mentioned operations, it could do various malicious activities such as altering the data, introducing arbitrary value during message transmission, refusing in protocol participation, illegally aborting the protocol etc [6-7].

Authors in this paper deals with Privacy Preserving k mean clustering on semi-honest model for secure mining. Pailler encryption scheme has been used for concealing of sensitive data from the various associated parties as well as from the outside attackers. Its homomorphism properties have been exploited for concealing sensitive information as well as maintaining data utility during entire operation.

Kantarcioglu et al. [8] focuses on association rule mining on horizontally partitioned data. The datasets are encrypted and transmitted to adjacent parties for further computations. Giannotti et al. [9] deals with outsourcing the encrypted data to the cloud server for privacy preserving association rule mining. It makes the use of Rob Fugal encryption scheme based on 1-1 substitution cipher. It is added in the fake transaction so to exhibit the same frequency as >= k-1. Li Weng et al. [10] proposes the framework including client, user and distrusted server. This system model relies on hashing and symmetric encryption scheme. Data owner computes hash of the message and partially encrypt it before outsourcing it to the mistrusted server. Client performs the similarity search by ranking the most similar candidates received from the server. One way hashing and encryption prevents the mistrusted server from guessing the correct data. Lin Zhang et al. [11] proposed decision tree mining based on differential privacy-protection mechanism. An efficient classifier is used to perturb the data by adding noise of either Laplacian or exponential mechanism based according to the user feedback. Different split solutions for continuous and discrete set are provided to reduce the error rate and optimizing the search scheme. Vidya et al. [12] improves the existing Random Decision Tree (RDT) for parallel and fully distributed data mining architecture.

Prominent research works has been done in the field of Privacy Preserving k-means clustering. K. Samanthula et al. [13] deals with k nearest neighbor classification for transporting encrypted relational data on the cloud server. The proposed model used Pailler encryption scheme for semi-honest as well as malicious model. It proposed additionally four arithmetic operations – Secure Multiplication, secure square Euclidean distance computation and secure bit decomposition. Query is submitted by the third party user and the data is classified accordingly. The protocol performance is evaluated for different parameter settings. Michal and Mathew [14] cluster the online data stream into large number of micro clusters. DBSTREAM it captures the density between micro clusters via shared density graph. This density information between micro-clusters is exploited during data reconstruction. It improves clustering quality while creating large number of micro clusters for achieving comparable results.

Outsourced encrypted mining using Pailler encryption scheme has been proposed by Ximeng L. et al. [15]. A trusted authority is present for key distribution. Encrypted datasets is transmitted to the cloud server for performing privacy preserving naive bayes classification on medical datasets in semi-honest model. Zhan Q et al. [16] propose privacy preserving frequent visual pattern mining of graphical data on cloud. It aggregates the summary over individual frame while concealing statistical sensitive information. The proposed algorithm properly optimize privacy budget over different stages in Frequent Pattern Mining and minimize data distortion. Yi-Ting-Cheng et al. [17] proposed sequential pattern mining of health care data by building classification model. The proposed framework consists of four stages such as data preprocessing, risk patterns mining, classification model and post analysis for early assessment of chronic disease. Chao Y H et al. [18] deals with medical image feature extraction by encryption using Pailler Homomorphic Cryptosystem for honest but curious model. Pailler Cryptosystem does not show multiplicative homomporphic encryption hence it cannot be used for medical image feature extraction. Peter S.W et al. [19] deals with privacy preserving distributed karnel based data mining for protection of karnel data from inside attackers. QJ Zhou et al. [20] proposed privacy preserving secure data aggregation in smart grid using Eigamal encryption scheme. It support additive homomorphic encryption scheme and hence cannot used in dynamic text mining and image feature extraction. This feature on medical datasets on the cloud server has been proposed using one-way trapdoor function by Zhou et al. [21]. Jerry Chung et al. [22] proposed data mining which try to reduce complexity using anonimization technique. Yagacharan Rahulamathawan et al. [23] deal with outsourcing clinical datasets in encrypted form for SVM classification. Other works based on SVM classification has also been discussed in [24-26]. Keke Chan et al. [27] deals with privacy preserving geometric perturbation for maintaining data correlations while reducing its dimension among multiple parties. Research works on [28-29] deals with outsourcing the encrypted data to the cloud server. Ming Li et al. [30] provide framework for storing and accessing patient centric data stored in the cloud server in a privacy preserving manner. Rongxing L. et al. [31] perform bayessian classification and diagnosis of heart disease based on patient history, physical examination and cathetirization result.

Authors in this paper propose PPkDC (Privacy Preserving k-means data clustering) approach on semi-honest model. The data is encrypted using Pailler homomorphic cryptosystem scheme and k-means clustering approach is applied on encrypted datasets. Encryption and permutation of cipher texts makes the cloud server even much harder in deciphering the input data of an organization.

The rest of the paper is organized as follows. In Section III, we discussed some preliminaries, which serve as the basis for the proposed work. Section IV discusses the system models and design goals. Section V discusses the framework. Section VI and VII discusses the security algorithm and it's UML Modeling. Finally in our last section, authors have discussed the conclusion and its further research directions.

2. SYSTEM ARCHITECTURE AND SECRITY REQUIREMENTS

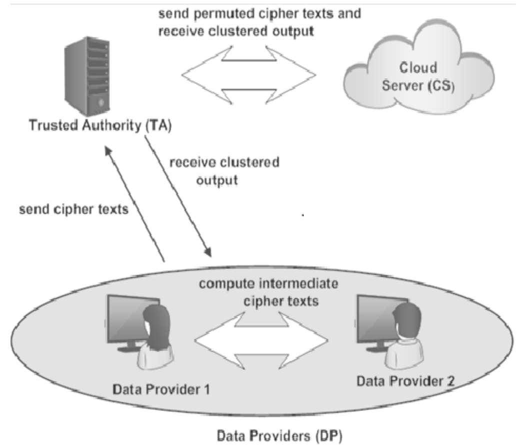

Figure 1. System Model Of Outsourcing Data To Cloud Server

The main motive in the proposed framework is to securely cluster the data owner's datasets for classifying datasets without leaking any private information to the malicious cloud server as well as to the various unauthorized parties associated with it.

2.1. System Architecture

Figure 1 illustrates the system model of the proposed PPkDC approach. It includes the following three entities: - Trusted Authority (TA), Cloud Server (CS) and Data Providers (DP). The overall framework of the PPkDC system is discussed below:

1. Trusted Authority (TA): Trusted Authority is trusted by all Data Providers in the framework and has the task of generating, distributing and managing key pairs to all participating parties associated in the system. It also performs other operations such as computing and communicating Hash values for verifying data integrity.

2. Data Provider (DP): Data Provider contains sensitive information which they wants to be mined without exposing it to unauthorized parties. These datasets are encrypted and transported via Trusted Authority to the mistrusted cloud server for performing secure data mining operation.

3. Cloud Server (CS*):* Cloud Server (CS) have unlimited storage space, processing and computation capabilities. Data Provider outsource their datasets to the Cloud Server and it stores, manages, train the datasets and send the computed results to the Data Provider via Trusted Authority. Cloud Server Administrator is being paid in return for using its services.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.3. Design Goals

In order to securely train the student's performance datasets, the proposed system should fulfill the following requirements:

(i) The proposed system should achieve the strong privacy: The exposing of sensitive personal data to the unsecured cloud platform is against the legal constraints. It will let the participating parties to unwillingly provide private data which could be serious threat to their privacy. Each Data provider's sensitive datasets should be protected from external parties as well as from the malicious Cloud Server (CS) during the entire process.

(ii) The clustering output should be accurate: The accuracy of the clustered output should not be compromised due to heavy computations on encrypted datasets.

(iii) The proposed system should be computationally efficient: The entire outsourcing and mining operation involves huge computation on encrypted datasets. In order to be practically efficient, the entire operation should be computationally feasible and operate in real time.

Table 1. Definitions and Notation

S.No.	Symbols	Definitions
1.	$x_{i1},x_{i2},x_{i3},............,x_{in}$	Plaintext data of data provider i
2.	$d[]=[d_{12},d_{13},.......,d_{ij}],i\neq j$	Array of intermediate cipher text computed between each tuple pair i and j, $i\neq j$
3.	$D_{12},D_{13},........,D_{ij}$	Permuted ciphertext data
4.	$H_{TA}(E_{pk}\,X_{ij}[])$	Hash value of the permuted ciphertext computed by TA
5.	$H_{CS}\,(E_{pk}\,X_{ijpt}[])$	Hash value of the final Clustered Ciphertext computed by CS
6.	$C_1^{\,t},C_2^{\,t},C_3^{\,t},......,C_k^{\,t}$	K-clustered ciphertext data at t^{th} iteration

3. PRIVACY PRESERVING PRELIMINARIES

In this section, authors have reviewed Pailler encryption scheme [32-33], k-mean clustering [34-35] and Hash value computation [36], which forms the basis of our proposed work. Table 1 lists some of the main notation to be frequently used in the present work.

3.1. Pailler Homomorphic Encryption

It is a probabilistic public-key algorithm for performing homomorphic operations on encrypted datasets [33]. It includes the following three steps:

(i) Key generation: Let p and q be the two large prime numbers of equal length chosen randomly and independently of each other such that

$$\gcd(pq,(p-1)(q-1))=1$$
$$|p|=|q|=l$$

(a) Compute $N=p*q$

(b) Choose random integer g where

$$g \in Z_N{}^2.$$

(c) Check the existence of the following multiplicative inverse μ to ensures n divides the order of g,

$$\mu=(L(g^{\lambda}\,(\mathrm{mod}\ N^2)))^{-1}\,(\mathrm{mod}\ N)$$

where , $\lambda=\mathrm{lcm}(p-1,q-1)$,

$$L(u)=(u-1)/N$$

(d) Generated public key $p_k\,(N, g)$ and secret key $s_k\,(\lambda\,,\mu)$ and send it to the receivers.

(ii) Encryption: To encrypt the message $m \in Z_N$, randomly choose a number $r \in Z_N$. Cipher text c of the message is computed as

$$c=g^m r^n\ (\mathrm{mod}\ N^2).$$

(iii) Decryption: For the cipher text c to decrypt where $c \in Z_N{}^{2*}$, the plaintext message m be computed as

$$m=L(c^{\lambda}\,(\mathrm{mod}\ N^2)).\ \mu\ (\mathrm{mod}\ N)$$

Pailler Homomorphic encryption scheme exhibits following three properties [15].

(i) Homomorphic Addition
(ii) Self-Blinding
(iii) Scalar Homomorphic Multiplication

(i) Homomorphic Addition: Multiplication of two encrypted cipher text message results in addition of original plaintext mod n.

Let m_1 and m_2 be the two message shared by mistrusted parties each having public key p_k and D_{sk} be the decryption function with the secret key s_k holds by any one party, then,

$$D_{sk}\,[E_{pk}(m_1).E_{pk}(m_2)\mathrm{mod}\ N^2]=(m_1+m_2)\mathrm{mod}\ N^2 \quad \text{–Equation (1)}$$

(ii) Self-Blinding: For a given cipertext E(x) it's plaintext cloud be obtain be computing

$$E_{pk}(-x) = E_{pk}(x)^{N-1} \quad \text{–Equation (2)}$$

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(iii) Scalar homomorphic Multiplication: Constant Multiplication of the original plaintext is derived by raising the cipher text to the constant power.

$D_{sk}[E_{pk}(m)^k \bmod N^2]=k*m.\bmod N^2 \quad k \in Z_N$ –Equation (3)

3.2. Secure Euclidean Distance Computation (SEDC):

Let A and B be the two parties each shares the public and private key p_k and s_k respectively and having inputs values (x_1, x_2) and (y_1,y_2) respectively. The goal is to securely compute $E_{pk}(\sqrt{(x_1-y_1)^2+(x_2-y_2)^2})$ by the Trusted third party such that that any of parties could not learn the input value of each other. Pailler homomorphic only support the limited homomorphic operation. Similar to SM Protocol (Secure Multiplication Protocol) [15] author here have discuss the successive steps for achieving Secure Euclidean Distance Computation. A and B each choose a secret random number r_x, and $r_y \in Z_N$ respectively, perform the calculations and outsource it to Trusted Third Party for performing further calculations which have the risks of exposure of sensitive data (As Each hold the decryption key s_k). Assume that each of them does not collude, B encrypt the values $E_{pk}(y_1)$, $E_{pk}(r_y)$ and perform the following computations.

$a = E_{pk}(y_1)*E_{pk}(r_y)$
$=E_{pk}(y_1+r_y)$ [According to Equation 1]
$a' = E_{pk}(r_y)$

These value a and a' are transported to A and Trusted Third Party respectively. The main reason for outsourcing a' to Third Party instead of A is that A possessing the decryption key may decipher the random value $E_{pk}(r_y)$. A performs the following computations

$b = E_{pk}(x_1)$
$b' = E_{pk}(r_x)$
$c = b*b'$
$= E_{pk}(x_1)* E_{pk}(r_x)$ [According to Equation 1]
$= E_{pk}(x_1+r_x)$

$e = D_{sk}(a)D_{sk}(c)$
$= D_{sk}[E_{pk}(y_1+r_y)]. D_{sk}[E_{pk}(x_1+r_x)]$
$= (y_1+r_y)(x_1+r_x)$

$E_{pk}(r_y)$ are transported by B to Trusted Third Party. It performs the following computations

$d = a'^{N-rx}$
$=E_{pk}(r_y)^{N-rx} \quad =E_{pk}(-r_y r_x)$

The number $N-r_x$, $E_{pk}(e)$, and b, $N-r_y$ are transported by A and B respectively to the Trusted Third Party, and it perform the following computation

$c'=E_{pk}(x_1)^{N-ry}$
$= E_{pk}(-x_1 r_y)$
$c''=E_{pk}(y_1)^{N-rx}$
$= E_{pk}(-y_1 r_x)$

$e' = E_{pk}(e) * c' * c'' *d$
$=E_{pk}[(y_1+r_y)(x_1+r_x)]* E_{pk}(-r_x * x_1)*E_{pk}(-y_1 * r_y)*E(-r_y . r_x)]$
$= E_{pk}(x_1y_1)$
$f=(D_{sk}(e'))$
$=(D_{sk}* E_{pk}(x_1y_1))$
$= x_1y_1$
$f'=E_{pk}(f)^{N-2}$
$= E_{pk}(x_1y_1)^{N-2}$

Similar to the above $d' =E_{pk}(x_2y_2)^{N-2}$ would be calculated

Values computed by A, $l =E_{pk}(x_1^2)$, $b= E_{pk}(x_2^2)$
$m= l *b$
$m= E_{pk}(x_1^2) * E_{pk}(x_2^2)$
$= E_{pk}(x_1^2 + x_2^2)$

Values computed by B, $l'=E_{pk}(y_1^2)$, $b'= E_{pk}(y_2^2)$
$m'= l'*b'$
$m'= E_{pk}(y_1^2) * E_{pk}(y_2^2)$
$= E_{pk}(y_1^2 + y_2^2)$

These values m and m' are transported by A and B respectively to the Trusted Third Party.

Now, Euclidean distance (d_{ij}) between A and B

$d_{ij} =m*m'* f' *d'$
$= E_{pk}(x_1^2 + x_2^2) * E_{pk}(y_1^2 + y_2^2) * E_{pk}(x_1y_1)^{N-2} * E_{pk}(x_2y_2)^{N-2}$
$=E_{pk}(x_1^2 + x_2^2) * E_{pk}(y_1^2 + y_2^2) * E_{pk}(-2x_1y_1) * E_{pk}(-2x_2y_2)$
$=E_{pk}(x_1^2 + x_2^2 -2x_1y_1) * E_{pk}(y_1^2 + y_2^2 -2x_2y_2)$
$=E_{pk}[(x_1-y_1)^2+(x_2-y_2)^2]$

The Euclidian distance for each of these n participating parties' pairs can similarly be calculated. The detailed computation steps of its implementation over cloud server will be further explained in Section 4.2 and 4.5.

4. PROPOSED PPkDC PROTOCOL:

Authors have proposed a PPkDC algorithm for clustering large number of datasets. Let there be the s participating parties $DP_1,DP_2,\ldots,DP_s$ having the horizontally partitioned datasets $x_1(x_{11},x_{12},x_{13},\ldots,x_{1n}),x_2(x_{21},x_{22},\ldots x_{2n}),\ldots,x_m(x_{m1},x_{m2},\ldots,x_{mn})$ and each having n attributes. The goal is to securely train the encrypted datasets while concealing the sensitive information from external parties as well as from the malicious Cloud Server and still getting the accurate clustered output. The proposed method is based upon Pailler homomorphic encryption scheme which consists of the following five phases will be discussed below.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1. Key pair generation and distribution:

In this phase, TA should generate and publish the security parameters and Key pairs to each Data Provider for encrypting messages before outsourcing. The security parameters (r, g, Z_N) public/private key pair $p_k(N,g)$, $s_k(\lambda,\mu)$ values are broadcasted to each Data Providers after key generation and distribution phase.

Each Data Provider DP_i generates the Keypairs. The detailed steps for key pair generation are as follows:

a) Choose two prime numbers p,q of equal length and generator g of the cyclic group G of the order Z_N^2 such that gcd (pq,(p-1)(q-1))=1

b) Compute N=p.q ,

c) Choose a random integer g where g ε Z_N^2.

d) Generate a constant value μ such that
$\mu=(L(g^{\lambda}(\mathrm{mod} N^2)))^{-1}(\mathrm{mod} N)$
where L(u)=(u-1)/n and λ=lcm(p-1,q-1)
A public key $p_k(n,g)$ and secret key s_k (λ ,μ) will be generated.

e) Choose a random number r ϵ Z_N

f) It sends the random number and Key pairs to each Data Providers.

Table 2 illustrates the pseudo-code of Privacy preserving k-means clustering

Table2. Algorithm for Privacy Preserving Key pair generation

TA:
1.generate two prime numbers p and q of equal length such that
gcd(pq,(p-1)(q-1))=1
2. compute N ← p.q
3. choose generator g Ɛ Z_N^2
λ =lcm((p-1)(q-1))
4. calculate μ such that
$\mu=(L(g^{\lambda} (\mathrm{mod}\ N^2)))^{-1} (\mathrm{mod}\ N)$
5. Choose a random number r ϵ Z_N
6. **Broadcast** Pailler parameter (r, Z_N*) public key P_k(N, g), private key S_k(λ ,μ) to all DP_i

4.2. Privacy Preserving Intermediate Cipher Text Computation Between Participating Parties

This algorithm is run by DP. In this phase, each Data Providers performs various intermediate computations after encrypting data and outsource them to the Trusted Authority. The horizontally partitioned datasets $x_1(x_{11},x_{12},x_{13},\ldots,x_{1n})$, $x_2(x_{21},x_{22},\ldots x_{2n}),\ldots,x_m(x_{m1},x_{m2},\ldots,x_{mn})$ of data provider DP_1, DP_2 ,.....,DP_s each having n attributes are encrypted by public key p_k.

Let each data provider choose a random number r_y ϵ Z_N and encrypt the dataset x_{ij}

$r_y \in Z_N$ y=1,2..........,s

$E_{pk}(x_{ij})=g^{x_{ij}}r_y^N(\mathrm{mod} N^2)$ i=1,2,......., m
j=1,2,.......,n

Intermediate cipher text computations take place between each participating parties DP_u and DP_v. Let r_u and r_v be the two different random number chosen by the party DP_u and DP_v respectively such that r_u, r_v ∈ Z_N. Secure Euclidian distance computation between any two tuples (say l and p, 1<=l<p<m) of two different parties pair DP_u and DP_v each having attributes value x_{li} and x_{pi} respectively(as explained in section C) is calculated as explained below:

DP_v

$a_{pluv} = E_{pk}(x_{li})*E_{pk}(r_v)$
$= E_{pk}(x_{li}+r_v)$
$a'_{pluv} = E_{pk}(r_v)$

This value a_{pluv} is outsourced to the Party DP_u. a'_{pluv} are outsourced to the Trusted Third Party. It performs the following computations.

DP_u

$b_{pluv} = E_{pk}(x_{pi})$
$b'_{pluv} = E_{pk}(r_u)$
$c_{pluv} = E_{pk}(x_{pi})*E_{pk}(r_u)$
$= E_{pk}(x_{pi}+r_u)$

$e_{pluv}=D_{sk}(a)*D_{sk}(c)$
$=D_{sk}[E_{pk}(x_{li}+r_v)]D_{sk}[(E_{pk}(x_{pi}+r_u)]$
$= (x_{li}+r_v)(x_{pi}+r_u)$

c_{pluv} , b_{pluv} ,e_{pluv} and b'_{pluv} are outsourced to Trusted Third Party .

TA

Each of the encrypted values $E_{pk}(x_{pi})$, $E_{pk}(x_{li})$ and random secret values N-r_u and N-r_v cannot be outsourced to other party as rival party poses decryption key could easily decipher data. So, the data are outsourced to TA for performing the following computation

a'' are transported by DP_v to the Trusted Third Party. It performs the following computations

$d'_{pluv} = a''^{N-ru} =E_{pk}(r_v)^{N-ru}$
$=E_{pk}(-r_v r_u)$
$c'_{pluv} = E_{pk}(x_{pi})^{N-rv}$
$=E_{pk}(-r_v.x_{pi})$
$c''_{pluv} = E_{pk}(x_{li})^{N-ru}$
$=E_{pk}(-x_{li}.r_u)$
$d_{pluv} = a'^{N-ru}$
$=E_{pk}(r_v)^{N-ru} =E_{pk}(-r_v r_u)$

$e'_{pluv} = E_{pk}(e_{pluv}) * c'_{pluv} * c''_{pluv} *d_{pluv}$
$=E_{pk} [(x_{pi}+r_u)(x_{li}+r_v)]* E_{pk}(-r_u * x_{1i})*E_{pk}(-x_{pi}* r_v)*E(-r_u. r_v)]$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$= E_{pk}[(x_{pi}.\, x_{li} + r_u.x_{li} + x_{pi}.\, r_v - r_u.r_v - r_u.x_{1i} - x_{pi}.\, r_v - r_u.\, r_v)]$
$= E_{pk}(x_{li}.x_{pi})$

Now Compute $f_{pluv}=(D_{sk}(e'_{pluv}))$
$=(D_{sk} * E_{pk}(x_{li}.x_{pi}))$
$= x_{li}x_{pi}$
$f'_{pluv}=E_{pk}(f_{pluv})^{N-2}$
$= E_{pk}(x_{li} * x_{pi})^{N-2}$

Computation performed by DP_u

$s'_{puv} = \prod_{j=1}^{n} Epk(x_{pj}^2)$
$= E_{pk}(x_{p1}^2)*E_{pk}(x_{p2}^2)*\ldots*E_{pk}(x_{pn}^2)$
$= E_{pk}(\sum_{j=1}^{n} x_{pj}^2)$

Computation performed by DP_v

$s'_{luv} = \prod_{j=1}^{n} Epk(x_{lj}^2)$
$=E_{pk}(x_{l1}^2)*E_{pk}(x_{l2}^2)*\ldots*E_{pk}(x_{ln}^2)$
$= E_{pk}(\sum_{j=1}^{n} x_{lj}^2)$

Outsource s'_{puv} and s'_{luv} to Trusted Authority

Following computations are performed by the Trusted Authority

$s'_{pluv} = \prod_{j=1}^{n} f'$
$= \prod_{j=1}^{n} E_{pk}(x_{pj} * x_{lj})^{N-2}$
$= E_{pk}[(-2)\sum_{j=1}^{n}(x_{pj} * x_{lj})]$

$s'_{SEDCpluv} = s'_{puv} * s'_{luv} * s'_{pluv}$
$=(\prod_{j=1}^{n} E_{pk}((x_{pj}^2) * E_{pk}(x_{lj}^2)*E_{pk}(x_{pj}*x_{lj})^{N-2}))$

$= (x_{p1}^2 - x_{l1}^2) + (x_{p2}^2 - x_{l2}^2) + (x_{p2}^2 - x_{l2}^2) + \ldots\ldots\ldots\ldots(x_{pn}^2 - x_{ln}^2)$
$=E_{pk}[(x_{p1}^2 - x_{l1}^2) + (x_{p2}^2 - x_{l2}^2) + \ldots\ldots + (x_{pn}^2 - x_{ln}^2)]$

$= \sum_{j=1}^{n} E_{pk}[(x_{pj} - x_{lj})^2]$, $1 \le p < l \le m, l \ne m$
$d_{pluv} = \sqrt{s'_{SEDCpluv}}$

These value d_{pluv} between each tuple values pair pl is outsourced to the Trusted Authority. For simplicity, we consider d_{pluv} as d_{pl}, which is the distance between any two tuple pair in a particular datasets.

These encrypted cipher text $E_{pk}(x_{ij}[])$ along with relative distances d_{pl} [] of each Data Provider pairs are permuted before outsourcing them to the Trusted Authority. Table 3 illustrates the algorithmic procedure to achieve Privacy preserving intermediate ciphertext computations.

Table 3. Algorithm for Privacy Preserving intermediate cipher texts computation

Input: $x_1(x_{11},x_{12},x_{13},\ldots,x_{1n}), x_2(x_{21},x_{22},\ldots,x_{2n}),\ldots\ldots\ldots, x_p(x_{p1},x_{p2},\ldots,x_{pn}), x_l(x_{l1},x_{l2},\ldots,x_{ln}),\ldots,x_m(x_{m1},x_{m2},\ldots,x_{mn})$

Output: $d_{pluv}[]$

1.**for** each parties DP_u and DP_v having datasets of n attributes(i=1,2...,n) **do**
2.choose random number
$r_y \epsilon Z_N$ y=1,2..........,s
3.$E_{pk}(x_{ij})=g^{xij}r_y^N(\mathrm{mod}N^2)$ i=1,2,.........,m, j=1,2,.........,n

DP_v:
1.choose random number $r_v \in Z_N$
2. compute $a_{pluv}=E_{pk}(x_{li})*E_{pk}(r_v)$
$= E_{pk}(x_{li}+r_v)$
3. $a''_{pluv} = E_{pk}(r_v)$
4. **return** a_{pluv} to DP_u, a''_{pluv}, $N-r_v$, $E_{pk}(r_v)$ to TA.

DP_u
1.choose random number $r_u \in Z_N$
2. $b_{pluv} = E_{pk}(x_{pi})$
3. $b'_{pluv} = E_{pk}(r_u)$
4. $c_{pluv} = E_{pk}(x_{pi})*E_{pk}(r_u)$
$= E_{pk}(x_{pi}+r_u)$
5. $e_{pluv}=D_{sk}(a)*D_{sk}(c)$
6. $= (x_{li}+r_v)(x_{pi}+r_u)$
7. **return** $N-r_v$, c_{pluv}, b_{pluv}, e_{pluv} and b'_{pluv} to TA and d_{pluv} to DP_v

TA
1. compute $d'_{pluv} = a'^{N-ru} = E_{pk}(r_v)^{N-ru}$
2. $=E_{pk}(-r_v\, r_u)$
3. $c'_{pluv} = E_{pk}(x_{pi})^{N-rv}$
4. $=E_{pk}(-r_v.x_{pi})$
5. $c''_{pluv} = E_{pk}(x_{li})^{N-ru}$
6. $=E_{pk}(-x_{li}.\, r_u)$
7. $d'_{pluv} = a'^{N-ru}$
8. $=E_{pk}(r_v)^{N-ru}$ $=E_{pk}(-r_v\, r_u)$
9. $e'_{pluv} = E_{pk}(e_{pluv}) * c'_{pluv} * c''_{pluv} * d_{pluv}$
$= E_{pk}(x_{li}.x_{pi})$
10. $f_{pluv} = =(D_{sk}(e'_{pluv}))$
$= x_{li}x_{pi}$
11. $f'_{pluv}=E_{pk}(f_{pluv})^{N-2}$
$= E_{pk}(x_{li}x_{pi})^{N-2}$

DP_u
1.$s'_{puv} = \prod_{j=1}^{n} Epk(x_{pj}^2)$
$= E_{pk}(\sum_{j=1}^{n} x_{pj}^2)$
2. outsource s'_{puv} to TA

DP_v
1. $s'_{luv} = \prod_{j=1}^{n} Epk(x_{lj}^2)$
$= E_{pk}(\sum_{j=1}^{n} x_{lj}^2)$
2. outsource s'_{luv} to TA

TA
1.$s'_{pluv} = \prod_{j=1}^{n} f'$
$= \prod_{j=1}^{n} E_{pk}(x_{pj} * x_{lj})^{N-2}$

$=E_{pk}[(-2)\sum_{j=1}^{n}(x_{pj} * x_{lj})]$
2. $s'_{SEDCpluv} = s'_{puv} * s'_{luv} * s'_{pluv}$
$=(\prod_{j=1}^{n} E_{pk}((x_{pj}^2) * E_{pk}(x_{lj}^2) * E_{pk}(x_{pj} * x_{lj})^{N-2}))$
$= \sum_{j=1}^{n} E_{pk}[(x_{pj} - x_{lj})^2]$, $1 \le p < l \le m, p \ne l$
3. $d_{pluv} = \sqrt{s'_{SEDCpluv}}$
4. **send** $E_{pk}(x_{ij})$ TA.

4.3. Privacy Preserving Permutation Of The Received Cipher Text

The encryption of the sensitive data is not sufficient. The malicious Cloud Server are interested in learning the sensitive data of each data provider, could still try to infer it by the relative ordering of cipher text. In order to prevent it, the encrypted coordinates array of each party are permuted by the Trusted Authority (TA) by the random permutation function π where

$E_{pk}(X_i[])=\pi(E_{pk}(x_i[]))$
$D_{pl}=\pi(d_{12},d_{13},d_{23}...,d_{pl})$ $1<=p<l<m$
$D_{pl}=\pi([d_{pl}[])$ $1 \le p < l \le m, p \ne l, lp \ne pl$

Hash value $H_{TA}(E_{pk}(X_{ij}[])$ of each permuted cipher texts $E_{pk}(X_{11}),E_{pk}(X_{12}),......,E_{pk}(X_{mn})$ are computed. These permuted encrypted cipher texts distance pairs $((E_{pk}(X_{mn}[]), D_{pl})$ are send to the Cloud Server. Table 4 illustrates the algorithmic procedure to achieve ciphertext and distance value permutation and hash value computation of the permuted ciphertext.

Table 4. Algorithm for Privacy Preserving permutation of the received cipher text

TA
Input: ciphertext cordinates $E_{pk}(x_{ij}[])$,distance pair ciphertext d_{pl}
Output: permuted ciphertext coordinate $E_{pk}(X_{ij}[],D_{pl}[])$, Hash Value $H(E_{pk}(X_{ij}[])$

1. receive $E_{pk}(x_{11}),E_{pk}(x_{12}),E_{pk}(x_{21})........,E_{pk}(x_{mn})$ from each parties
2. receive $d_{12},d_{13},d_{23}, ...d_{pl}$ from each p-l pair, p≠l
3. **for** (i=1,...,m) **do**
4. $D_{pl}=\pi(d_{12},d_{13},d_{23}...,d_{pl})$ $1<=p<l\le m, l \ne p, lp \ne pl$
5. $D_{pl}=\pi([d_{pl}[])$
6. **for** (i=1,2,….m) **do**
7. $E_{pk}(X_i[])=\pi(E_{pk}(x_i)$
8. compute Hash $H_{TA}(E_{pk}(X_{ij}[])$
9. **end for**
10. **return** $E_{pk}(X_i),D_{pl}[]$ to CS

4.4. Privacy preserving K-means clustering on Outsourced Data

The main goal in this phase is to build secure clusters on trained datasets without revealing the actual data to the malicious Cloud Server. As k-means clustering on encrypted datasets involves several iterations and often requires decryption key to achieve it, the datasets are outsourced to Trusted Authority with each iteration for achieving secure Euclidian distance computation until the mean value becomes constant. Cloud Server receives the encrypted permuted datasets D_{pl} along with $E_{pk}(X_{ij}[])$ from Trusted Authority. It performs Euclidean distance comparison (as explained in section 2) between each tuple coordinates and encrypted centroid to construct secure clustered model. Cloud Server arbitrarily chooses any k random points for training encrypted datasets column $E_{pk}(X_i[])$. Eucledian Distance comparison between each datasets column are performed and Temporary clusters $C_1^1, C_2^1,..., C_k^1$ are created based on the outsourced distance at the first iteration. Assume that each clusters values has minimum z tuples (1≤z≤m) (or say, z rows and n column), mean value $E_{pk}(\bar{x}_{jpt})$ of each cluster p at the t^{th} iteration are calculated by the following formula:

$E_{pk}(\bar{x}_{jpt}) = \frac{1}{z}[\prod_{i=1}^{z} E_{pk}(X_{ijpt})]$
$=\frac{1}{z}[E_{pk}(X_{1jpt}) * E_{pk}(X_{2jpt}) * E_{pk}(X_{3jpt}) * ... * E_{pk}(X_{zjpt})]$

$E_{pk}(\bar{x}_{jpt}) = \frac{1}{z} E_{pk}(\sum_{i=1}^{z} X_{ijpt})$ (where $1 \le p \le k$)

At first iteration, t=1
$E_{pk}(\bar{x}_{jp1}) = \frac{1}{z}[\prod_{i=1}^{z} E_{pk}(X_{ijp1})]$
$=\frac{1}{z}[E_{pk}(X_{1jp1}) * E_{pk}(X_{2jp1}) * E_{pk}(X_{3jp1}) * .. * E_{pk}(X_{zjp1})$
]
$= \frac{1}{z}[E_{pk}(X_{1jp1} + X_{2jp1} + X_{3jp1} + ...+ X_{zjp1})]$
$= \frac{1}{z}(E_{pk} \sum_{i=1}^{z} X_{ijp1})$

$E_{pk}(X_{ijpt})$ is the i^{th} row and j^{th} column of a particular tuple associated with any cluster p and t^{th} iteration.
Euclidean distance computation on the encrypted coordinates could not be possible without decryption key. The revealing of decryption key to insecure Cloud Server could be risky, so mean values $E_{pk}((\bar{x}_{jpt})$ along with clustered coordinates $C_1^1,C_2^1,...,C_k^1$ are outsourced to the Trusted Authority for Euclidean distance computation at each iteration.
Euclidean distance between the mean values $E_{pk}((\bar{x}_{jpt})$ and each clustered coordinates $E_{pk}(X_{ijpt})$ are calculated by the formula as mentioned below .
Sum of squares s_{ijpt} of any row $E_{pk}(X_i)$ of the cluster p at t^{th} iteration

$s_{ijpt} = \prod_{j=1}^{n} E_{pk}((D_{sk} * E_{pk}(X_{ijpt})^2))$

$=E_{pk}((D_{sk}*E_{pk}(X_{ijpt})^2))*E_{pk}((D_{sk}*E_{pk}(X_{ijpt}))^2)*E_{pk}((D_{sk}*E_{pk}(X_{ijpt}))^2).......E_{pk}((D_{sk} * E_{pk}(X_{inpt}))^2)]$
$=E_{pk}(X_{i1pt})^2 * E_{pk}(X_{i2pt})^2 * E_{pk}(X_{i3pt})^{2} * .... * E_{pk}(X_{inpt})^2$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$=E_{pk}[X_{i1pt}^2+X_{i2pt}^2+..................+X_{inpt}^2]$
$= E_{pk}(\sum_{j=1}^{n} X_{ijpt}^2)$

Similarly, sum of squares of mean value s_{mpt} of any cluster p at any t^{th} iteration

$s_{mpt} = [\prod_{j=1}^{n} E_{pk}(\bar{x}_{jpt}^2)]$
$=E_{pk}(D_{sk}*E_{pk}(\bar{x}_{1pt}))^2*E_{pk}(D_{sk}*E_p(\bar{x}_{2pt}))^2* E_{pk}(D_{sk}*E_{pk}(\bar{x}_{3pt}))^2).......*E_{pk}(D_{sk}*E_{pk}(\bar{x}_{zpt}))^2)$
$=E_{pk}(\bar{x}_{1pt}^2)*E_{pk}(\bar{x}_{2pt}^2)*E_{pk}(\bar{x}_{3pt}^2)......*E_{pk}(\bar{x}_{zp}^2)$
$=E_{pk}[(\bar{x}_{1pt}^2)+(\bar{x}_{2pt}^2)+(\bar{x}_{3pt}^2)......+(\bar{x}_{npt}^2)]$
$=E_{pk}(\sum_{j=1}^{n} \bar{x}_{jpt}^2)$

t=number of cluster iterations
i=number of rows
j=number of column
p= clusters index (1<p<=k)
$E_{pk}(\bar{x}_{jpt})$ = encrypted mean values for j^{th} column of p^{th} cluster at t^{th} iteration
Let choose k $\mathcal{E}$ Z_N

$h=E_{pk}(D_{sk}(E_{pk}(X_{ijpt})E_{pk}(k)).............D_{sk}\{(E_{pk}(\bar{x}_{jpt}) E_{pk}(k)\}*\{[E_{pk}(X_{ijpt})^k]^{N-1}\}* E_{pk}[(\bar{x}_{jpt})^k]^{N-1}*E_{pk}(k^2)^{N-1}$

$=E_{pk}(D_{sk}(E_{pk}(X_{ijpt}+k).D_{sk}(E_{pk}(\bar{x}_{jpt}+k))*E_{pk}(-kX_{ijpt})*E_{pk}(-k\bar{x}_{jpt})*E_{pk}(-k^2)$

$[E_{pk}(X_{ijpt})^k]^{N-1}=E_{pk}(X_{ijpt})^{-k}]$[According to Equation (2) and (3)]
$[E_{pk}(k^2)^{N-1} = E_{pk}(-k^2)]$ [According to Equation (2)]

$=E_{pk}[(X_{ijpt}+k)(\bar{x}_{jpt}+k)]*E_{pk}(-kX_{ijpt})*E_{pk}(-k\bar{x}_{jpt})*E(-k^2)$

$h=E_{pk}(X_{ijpt}\,\bar{x}_{jpt})$

$h' = D_{sk}(h)$
$=D_{sk}(E_{pk}(X_{ijpt}\,\bar{x}_{jpt}))$
$=(X_{ijpt}\,\bar{x}_{jpt})$

$q=E_{pk}(h')^{N-2}$
$=E_{pk}(X_{ijpt}\,\bar{x}_{jpt})^{N-2}$

$s_{ijmpt} = \prod_{j=1}^{n} E_{pk}(X_{ijpt} * \bar{x}_{jpt})^{N-2}$
$=E_{pk}((2X_{i1pt} * \bar{x}_{1pt})*(2X_{i2pt} * \bar{x}_{2pt})*.................*(2X_{inpt} * \bar{x}_{npt}))$

Now Secure Euclidean Distance Computation (SEDC) of between any data point $E_{pk}(x_{ijpt})$ and the centroid $E_{pk}(\bar{x}_{jpt})$ of any cluster p at t^{th} iteration is given by:

$s_{SEDCt} = s_{ijpt} * s_{mpt} * s_{ijmpt}$
$s_{SEDCt}=$
$\prod_{j=1}^{n}[E_{pk}(X_{ijpt}^2)*E_{pk}(\bar{x}_{jpt}^2)*E_{pk}(X_{ijpt}*\bar{x}_{jpt})^{N-2}]$
$=\prod_{j=1}^{n}[E_{pk}(X_{ijpt}^2)*E_{pk}(\bar{x}_{jpt}^2)*E_{pk}(-2)(X_{ijpt}*\bar{x}_{jpt})]$
$=E_{pk}\sum_{j=1}^{n}(X_{ijpt}^2 + \bar{x}_{jpt}^2 -2*X_{ijpt}*\bar{x}_{jpt})]$
$= E_{pk}(\sum_{j=1}^{n}(X_{ijpt} - \bar{x}_{jpt})^2)$

$D_{ijpt}=\sqrt{s_{SEDCt}}$

This is the Euclidean distance value of between any data point $E_{pk}(X_{ijpt})$ and the centroid $E_{pk}(\bar{x}_{jpt})$ of any cluster p. The computed distance values D_{ijpt} are outsourced to the Cloud Server. By taking mean value $E_{pk}(\bar{x}_{jpt})$ as the center and D_{ijpt} as distance with each coordinates, clusters are created at each iteration until its mean value $E_{pk}(\bar{x}_{jpt})$ becomes constant at some t^{th} iteration. Hash Value $H_{CS}[(E_{pk}(X_{ijpt})]$ of final cluster coordinates are computed and this value along with clustered output $C_1^t,C_2^t,...,C_k^t$ are outsourced to the Trusted Authority. Table 5 illustrates the algorithmic procedure to achieve Privacy Preserving k-means clustering.

Table 5. Algorithm for Privacy Preserving k-means clustering $E_{pk}(X_{ijpt}[])\leftarrow C_1^n,C_2^n,.....,C_k^n$

CS
Input: $E_{pk}(X_{ij})$, D_{pl} []
Output: k clusters $C_1^t,C_2^t,....,C_k^t$.
 	
Algorithm:
1. receive input array D[], $E_{pk}(X_{ij}[])$ from TA
2. Initialize C=φ, C_p^t = φ , A_{pt}[]= φ
3. $A_{k1}[] = rand[E_{pk}(X_i[])]$ p ϵ k
// arbitrary choose any k random columns or cluster points A_{k1} in $E_{pk}(X_i[])$
4. **for** (i=1 to m)
5. **for** (p=1,.....k) **do**
6. {
7. t=1 //for 1st iteration
8. {
9. **set** $E_{pk}(X_{ijp1})=E_{pk}(X_{ij}[])$
10.compute Eucledian Distance($A_{p1}[]$, $E_{pk}(X_i)$)
11. **if** (($A_{p1}[]$-E(X_r[])<$A_{p1}[]$-E(X_s[])) [r, s Є i]
12. ($D_{rp1} < D_{sp1}$)
13. {
14. $C_p^1 = E(X_r)$
15. }
16. **else**
17. {
18 . $C_p^1 = E(X_s)$
19. }
20.
21. (A_{p1}[]=Φ)
22.}
23. C=C_p^1 //create temporary clusters $C_1^1,C_2^1,....,C_k^1$ in 1st iteration
24. compute mean= $E_{pk}(\bar{x}_{jp1}) = \frac{1}{z}[\prod_{i=1}^{z} E_{pk}(X_{ijp1})$ (where 1 ≤ z ≤ m)
25. **if** (A_{p1} = $E_{pk}(\bar{x}_{ip1})$))
26. {
27. **return** $C_1^1,C_2^1,C_3^1,....,C_k^1$ as final cluster

28. to TA
29. }
30. **else**
31. {
32. t++;
33 goto sos
34. **receive** D_{ipt} from the CS
35. {
36. **for** (p=1,...k)
37. {
38. **if** ($D_{rpt} < D_{spt}$) [r,s ∈ i]
39. {
40. $C_p^t = E_{pk}(x_{rjpt})$ $[1 \le r < s \le m,\ r \ne s]$
41. **else**
42. $C_p^t = E_{pk}(x_{sjpt})$
43. }
44. *update* $C=C_P^t$
45. compute $E_{pk}(\overline{x}_{jpt}) = \frac{1}{z}[\prod_{j=1}^{n} E_{pk}$ (X_{ijpt})
46. } **while** ($D_{jpt}=\varphi$)
47. end **for**
48. }
49. } **while** $((E_{pk}(\overline{x}_{jp(t-1)})=E_{pk}(\overline{x}_{jpt}))$
50.consider temporary clusters $C_1^t, C_2^t, \ldots, C_k^t$ as final clusters
51.compute$H_{CS}(E_{pk}X_{ijpt}[])$
//compute the hash value of each ciphertext
52. return clusters $C_1^t, C_2^t, \ldots, C_k^t$ as the final clusters and $H_{CS}(E_{pk}(X_{ijpt}[]))$ to TA
53. }
54. }
end for
TA:
1.receive $C_1^1, C_2^1, C_3^1, \ldots, C_k^1$ from the CS
Sos:
1. Compute SEDC($E_{pk}(X_{ijp1})$, $E_{pk}(\overline{x}_{jp1})$) .
2.**return** D_{ijpt} to the CS

The algorithmic steps to achieve Secure Eucledian Distance Computation (SEDC) between each participating parties has been illustrated in Table 6.

Table 6. Algorithm for Secure Euclidean distance Computation (SEDC) between centroid $E_{pk}(\overline{x}_{jpt})$ and each clusters coordinates of $C_1^t, C_2^t, C_3^t, \ldots, C_k^t$.

1. **receive** C, $E_{pk}(X_{ijpt})$ values from CS
2. **for** (p=1,2,....k) //calculate between centroid and each cluster coordinates
3. {
4. compute $s_{ijpt} = \prod_{j=1}^{n} E_{pk}((D_{sk} * E_{pk}(X_{ijpt})^2))$

$= E_{pk}(D_{sk}(E_{pk}(X_{i1pt}))^2) * E_{pk}(D_{sk}(E_{pk}(X_{i2pt}))^2) * E_{pk}(D_{sk}(E_{pk}(X_{i3pt}))^2) \ldots\ldots. E_{pk}(D_{sk}(E_{pk}(X_{inpt}))^2)]$

$= E_{pk}(\sum_{j=1}^{n} X_{ijpt}^2)$

5. compute $s_{mpt} = \prod_{j=1}^{n} E_{pk}((D_{sk} * E_{pk}(\overline{x}_{jpt}))^2)$

$= E_{pk}(D_{sk}*E_{pk}(\overline{x}_{1pt}))^2 * E_{pk}(D_{sk}*E_{pk}(\overline{x}_{2pt}))^2 * E_{pk}(D_{sk}*E_{pk}(\overline{x}_{3pt}))^2) \ldots\ldots. * E_{pk}(D_{sk}*E_{pk}(\overline{x}_{jpt}))^2)$

$= E_{pk}(\sum_{j=1}^{n} \overline{x}_{jpt}^2)$

6.choose $k \in Z_N$

7.compute h=
$E_{pk}(D_{sk}(E_{pk}(X_{ijpt})E_{pk}(k)) \ldots\ldots\ldots D_{sk}\{(E_{pk}(\overline{x}_{jpt})E_{pk}(k)\} * \{[E_{pk}(X_{ijpt})^k]^{N-1}\} * E_{pk}[(\overline{x}_{jpt})^k]^{N-1} * E_{pk}(k^2)^{N-1}$
$= E_{pk}(X_{ijpt}\,\overline{x}_{jpt})$

8. $h' = D_{sk}(h)$
$= D_{sk}(E_{pk}(X_{ijpt}\,\overline{x}_{jpt}))$
$= (X_{ijpt}\,\overline{x}_{jpt})$

9. $q = E_{pk}(h')^{N-2}$
$= E_{pk}(X_{ijpt}\,\overline{x}_{jpt})^{N-2}$
10. $s_{ijmpt} = \prod_{j=1}^{n} E_{pk}(X_{ijpt} * \overline{x}_{jpt})^{N-2}$ //compute s_{ijmpt}
11. $s_{SEDCt} = s_{ijpt} * s_{mpt} * s_{ijmpt}$
12. $= \prod_{j=1}^{n} [E_{pk}(X_{ijpt}^2) * E_{pk}(\overline{x}_{jpt}^2) * E_{pk}(X_{ijpt} * \overline{x}_{jpt})^{N-2}]$
$= E_{pk}(\sum_{j=1}^{n}(X_{ijpt} - \overline{x}_{jpt})^2)$
13. calculate $D_{ijpt} = \sqrt{s_{SEDCt}}$
14. **return** D_{ijpt} to the CS.
15. end **for**

4.5. Privacy Preserving Secure Receiving Of Clustered Output

This phase concerns with receiving the clustered cipher texts from the Cloud Server and sending them to the Data Providers. Hash value computation is used for checking the integrity of received message from the Cloud Server.

The clustered data C_1^t, C_2^t, C_3^t ,.....,C_k^t along with $H_{CS}(E_{pk}(X_{ijpt}[])$ values are send to the Trusted Authority. It confirms integrity of the received cipher text by re computing Hash value and comparing them all with the original cipher text. $H_{CS}(E_{pk}(X_{ijpt}[]) = H_{TA}(E_{pk}X_{ij}[])$

It accepts the clustered output if the hashed value of original cipher text matches with the clustered output otherwise rejects the unmatched cipher text.

It request and receives the trained clustered datasets C_1^t, C_2^t, C_3^t,....., C_k^t. Table 7 depicts the algorithmic procedure for comparing hash value and outsourcing the authenticated clustered output to each participating parties.

Table 7. Algorithm for Outsourcing clustered output $C_1^t, C_2^t, \ldots, C_k^t$ to the participating parties

TA
1.receive clusters $C_1^t, C_2^t, \ldots, C_k^t$, $H_{CS}(E_{pk}(X_{ijpt}[])$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

from CS
for (i=1.....n) **do**
for (j=1,....m) **do**
compare ($H_{TA}(E_{pk}(X_{ij}[]) = H_{CS}(E_{pk}(X_{ijpt}[]))$
if ($(H_{TA}(E_{pk}(X_{ij}[]) = H_{CS}(E_{pk}(X_{ijpt}[]))= =1$) **then**
accept the clustered output
else
reject the clustered output
end for
end for
return clustered output $C_1^t, C_2^t, C_3^t, \ldots, C_k^t$ to DP

Authors have discussed the UML Modeling of the proposed framework. It includes three stakeholders- *Data Provider, Trusted Authority* and *Cloud Server*. Use case diagram illustrates how these stakeholders interact together to perform the entire operation. *Trusted Authority* has the tasks of generating, computing and publishing public parameters as well as public/private key pairs to all *Data Provider. Data Provider* has the responsibility of accepting public parameters and key pairs, encrypting data, computing intermediate cipher text values with Trusted Authority and sending it to the *Trusted Authority*. *Trusted Authority* has the responsibility of cipher text permutation, hash value computation and send it to *Cloud Server*. *Cloud Server* performs various intermediate computations together with *Trusted Authority* during k-mean clustering of the received cipher text. It also computes the hash value of the clustered output. This hash value along with clustered output is send to the *Trusted Authority*.

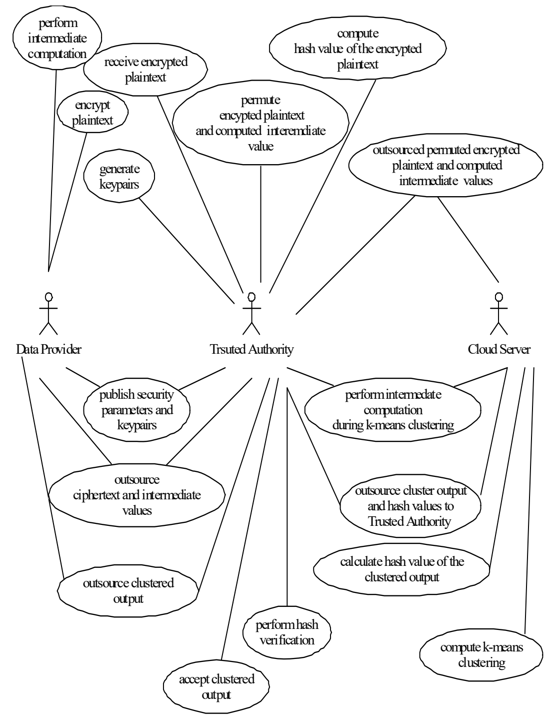

Figure 2. Use Case Diagram For Proposed Ppkdc Model

Trusted Authority has the responsibility of performing Hash verification and accepting clustered output. It then sends the clustered output to the Data Provider. Use case diagram of the entire operation has been illustrated in the figure 2 above.

5. ANALYSIS OF *PPkDC* MODEL

Authors have analyzed the proposed *PPkDC* protocol to determine whether it achieves the privacy requirements as mentioned in the above section III.

1. Security from external malicious adversaries

Hash verification scheme will aid the Trusted Authority to detect the lost, corrupted or unauthorized packet if any, introduced by the malicious cloud server. TA will abort the entire protocol whenever tempering in the clustered cipher texts output is detected

2. Privacy of Phase 2 of PPkDC

Security of the protocol totally depends upon the cryptographic PPkDC scheme. External eavesdroppers are unable to decode the sensitive data X_{ij} if they intercept it during its transmission route from DP to TA via CS. This is because each Data Provider's intermediate data are twofold encrypted- by the DP's public key $p_{k.}$ and the random number r. One has to access the DP's secret key s_k as well as random number r for deciphering the cipher texts. Data Provider does not collude, each of them could not learn the sensitive

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

information as they receive only encrypted values via Trusted Third Party during computation. Each Data Provider generates random value and the data exchanged with other participating parties are only the arithmetic operation on encrypted data and random numbers, they could not learn the original data by their decryption key.

3. Privacy in Phase 3, 4 and 5 of PPkDC

Permutation function π applied by TA in Phase 4.3 would prevent any malicious Cloud Server from deducing the corresponding plaintext of each Data Provider's. K-mean clustering on encrypted intermediate cipher text values ($E_{pk}(X_{ijpt})$) in Phase 4.4 would let the cloud server to learn only correlation among different data items without inferring real sensitive datasets $E_{pk}(X_{ij}[])$ of each Data Provider. Hash verification of the computed ciphertext prevents any malicious party to introduce any spurious, tempered or corrupted value in Phase 4.5.

6. RESULTS AND DISCUSSION

Authors have analyzed the efficiency of the algorithm in terms of computation and communication costs. Complexity of the proposed work has been compared with the existing approach. Complexity during key generation and distribution phase is O(s), where s is the total number of parties involves in computation. Total computation complexity of ciphertext encryption costs is O ($m_i n$), where m_i and n are the column and row respectively of each i^{th} participating parties. Total Eucledian distance computation complexity of intermediate ciphertext computation is $\sum_{i=1}^{m} n(m-i)$ i.e. of the order of O(mn),where m is the total number of column of the joint datasets of all participating parties. Total computation costs of ciphertext data permutation is of the order of O(1). Distance comparison complexity between each tuple is O ($\sum_{i=1}^{m}(m-1)$) i.e. of the order of O(m). Total mean value computation costs is O(k),where k is the number of clusters while Eucledian distance computation costs between each cluster coordinates and mean value is O(km).Total computation complexity during the t iterations is O[t(k+km)]. Overall complexity of the proposed algorithm includes complexity of each individual phase as mentioned in the research work. Protocol complexity is linearly depends on the data volumes of each participating parties, number of clusters values k and total number of iterations t. The complexity increases obviously with increase in data volumes which increase the computation costs and total number of iterations. The complexity of the algorithm is of the order of O(s), where s is the number of participating parties as compared to the order of s^2 by Jianming Zhu [37] in Privacy Preserving Collaborative data mining. Communication cost of the protocol in [43] is of the order O (n^2) where n is the number of each participating parties. Computation complexity in PPkDC approach varies linearly with the various parameters as compared to O(m^3) in [38] for two m*m matrix datasets during secure inverse of matrix sum protocol. Communication and computation costs of the algorithm is O(n^2k^2) in [39],where n is the number of participating parties and k is the neighboring points. The communication costs are O (n^3) for encryption for n data size [40]. The total comparison costs in Hong Rong et al.[41] approach is the order of O(d^2), where d is the number of fake tuples, compare to our O(m) distance comparison costs. For c clusters [42], requires c^2 cluster computation costs compared to the linear relation in our approach.

8. CONCLUSION

Heavy computation and storage space needed during mining will enable the participating parties having less computational power and resources to outsource their data on the Cloud Server. Authors have studied the problem of Privacy Preserving k-mean clustering of encrypted datasets outsourced to the cloud platform. Semi-honest model has been assumed where adversary are interested in learning the sensitive information during protocol execution. The method prevents external adversary from learning sensitive data during the course of transmission in the mining process. Authors have analyzed the efficiency of the algorithm in terms of complexity ,compared with the existing approach and is found that it is linearly dependent on the various parameter values hence is most efficient. Applying the same approach using zero-knowledge proof for malicious model could be a better future scope. Authors will design the UML model of the proposed system in its Object Oriented implementation. It will be helpful in implementing this approach for the other types of Data Mining algorithm also.

REFRENCES

[1] Dimitrios G K, Michael G X, and Charalampos Z. P," Cloud Federation and the Evolution of Cloud Computing. Computer", *Proc. IEEE*, 2016, pp. 96-99.

[2] Wahid H, Rao M L, Nazar A S, "Secure Agent Based Architecture for Resource Allocation in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Cloud Computing", *Proc*. IEEE , 2016, pp.1-6.

[3] Jianhao X., "Network Information Searching Technology Research based on Cloud Computing",*Proc. on Smart Grid and Electrical Automation. IEEE*, 2016 ,pp.132-135.

[4] Declan C, Kevin A J ,Khangelani V,."How Cloud Computing Influences Business Strategy within South African Enterprises", *Proc. on Emerging Technologies and Innovative Business Practices for the Transformation of Societies, IEEE*, *(EmergiTech)* 2016, pp. 272-278.

[5] Andrew C Y, "Protocol for secure computations", *23rd Annual Symposium on foundations of computer Science, IEEE*, 1982 :160-164.

[6] Ting W, Wenjun L ,"Design and Analysis of Private-Preserving Dot Product Protocol", *Proceedings on Electronic Computer Technology, IEEE Computer Society*, 2009,pp. 531-535.

[7] Rahena A, Rownak J C, Keita E,Tamzida I, Mohammad S R, Nusrat R., "Privacy-Preserving Two-Party k-Means Clustering in Malicious Model" , *Proc. 37th Annual Computer Software and Applications Conference Workshops. IEEE*, 2013, pp.121-126.

[8] Murat K and Chris C, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data" *,IEEE Transactions on knowledge and data engineering*, Vol. 16 ,No. 9, 2004, pp.1026-1037.

[9] Fosca G, Laks V S L, Anna M, Dino P, and Hui (Wendy) W.,"Privacy-Preserving Mining of Association Rules From Outsourced Transaction Databases, *IEEE Systems Journal*, Vol.7,No.3, 2013,pp. 385-395.

[10] Li W, Laurent A, Teddy F, "Privacy Preserving Outsourced media search", *IEEE Transactions on knowledge and Data Engineering*, Vol. 28, No.1, 2016, pp.2738-2751.

[11] Lin Z, Yan L, Ruchang W, Xiong F, Qiaomin L, "Efficient privacy-preserving construction model with differential privacy technology", *Journal of System Engineering and Electronics*, Vol.28, No.1, 2017,pp.170- 178.

[12] Jaydeep V, Basit S, Wei F, Danish M, David L, "Random Decision Tree framework for privacy-preserving data mining", *IEEE Transactions on dependable and secure Computing*, Vol. 11, No.5, 2014,pp. 399-411.

[13] Bharath K S, Yousef E, and Wei J, "k-Nearest Neighbor Classification over Semantically Secure Encrypted Relational Data", *IEEE Transactions on Knowledge and Data Engineering*, Vol.27, No.5, 2015, pp.1261-1273.

[14] Michal H, Mathew B , "Clustering Data Streams Based on Shared Density between Micro-Clusters", *IEEE Transactions on knowledge and Data Engineering*, Vol. 28,No.6, 2016,pp.1449 –1461.

[15] Ximeng L, Rongxing L, Jianfeng M, Le C, and Baodong Q, "Privacy-Preserving Patient-Centric Clinical Decision Support System on Na¨ıve Bayesian Classification", *IEEE Journal of Biomedical and health informatics*, Vol. 20,No.2, 2016,pp. 655-668.

[16] Zhan Q, Kui R,Ting Y,Jian W, "DPCode:Privacy Preserving frequent visual pattern publication on cloud", *IEEE Transactions on Multimedia*, Vol.18,No.5, 2016, pp.929-939.

[17] Yi T C, Yu F L, Kuo H C,Vincent S T, "Mining Sequential Risk pattern from large scale Clinical database for early assessment for chronic diseases: A Case Study on Chronic Obstructive Pulmonary Disease", *IEEE Journal of Biomedical and health informatics*, Vol.21, No.2, 2017,pp.303-311.

[18] Chao Y H, Chun S L, and Soo C P, "Image feature extraction in encrypted domain with privacy preserving SIFT, *IEEE Transaction on Image Processing*, Vol.21, No. 11, 2012,pp.4593–4607.

[19] Peter S W, Feipei L, Hsu-Chun H, and Ja-Ling W, "Insider Collusion Attack and Karnel based Privacy Preserving data mining systems", *Special Section on latest advances and emerging applications on data hiding,* Vol. 4, 2016,pp. 2244 – 2255.

[20] QJ. Zhou, X. Lin, X. Dong, and Z. Cao, "PSMPA: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributed m-healthcare cloud computing system", *IEEE Trans. Parallel Distributed System,* Vol.26,No.6 ,2015,pp.1693-1703.

[21] Jun Z, Zhenfu C, Xiaolei D, and Xiaodong L, "PPDM: A Privacy-Preserving Protocol for Cloud-Assisted e- Healthcare Systems", *IEEE Journal of selected topics in signal processing*, Vol.9, No.7, 2015, pp. 1332-1344.

[22] Jerry C W L, Quankun L, Philippe F V and Tzung P H," PTA: An Efficient System for

Transaction Database Anonymization", *IEEE Journal*, Vol: 4, 2016, pp.6467-6479.

[23] Yagacharan R, Suresh V, Raephel C W P, Jonathan A C, Mutthukrishnan R ,“ Privacy Preserving Decison Support system using gaussian karnel based classification”, *IEEE Journal of Biomedical and health information*, Vol.18, No.1, 2014, pp.56-66.

[24] K.Lin and M.Chen ,”On the design and analysis of the privacy-preserving SVM classifier” ,*IEEE Trans. Knowl. Data Eng*ineering, Vol.23, No.11, Nov. 2011,pp. 1704– 1717

[25] H. Yu, X. Jiang, and J. Vaidya, Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data, *Proc. ACM Symp. Appl. Comput*. 2006, pp. 603–610.

[26] H. Li, L. Xiong, L, Ohno-Machado, and X. Jiang, “Privacy preserving RBF kernel support vector machine”, *BioMed. Res. Int*., Vol. 2014, 2014.

[27] Keke C, Ling L, Privacy Preserving collaborative mining with geometric data perturbation, *IEEE Transactions on Parallel and Distributed Systems*, (Vol.20, No.12), 2009, pp.1764 – 1776.

[28] A. Abbas and S. U. Khan ,”A review on the state-of-the-art privacy preserving approaches in the e-health clouds”, *IEEE Journal of Biomedical and Health Informatics*, Vol.18, No.4, 2014, pp.1431–1441.

[29] Y. Tong, J. Sun, S. S. M. Chow, and P. Li, “Cloud-assisted mobile-access of health data with privacy and auditability”, *IEEE Journal of Biomedical and Health Informatics*, Vol. 18, No.2, 2014, pp. 419–429.

[30] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou, “Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption”, *IEEE Transactions on Parallel Distributed Systems*, Vol.24, No.1, 2013, pp. 131–143.

[31] Rongxing L, Hui Z, Ximeng L, Joseph K L, Jun S, “Towards efficient and privacy-preserving computing in big data era”, *IEEE Network*, Vol. 28,No.4, 2014, pp. 46–50.

[32] Xun Y ,Russell P, Elisa B, “Homomorphic Encryption and Applications”, *Springer Briefs in Computer Science,* Springer International Publishing, 2014, pp. 27-46.

[33] Pierre-Alain F, Guillaume P and Jacques, “Sharing Decryption in the Context of Voting or Lotteries, Financial Cryptography”, *Lecture Notes in Computer Science*, Springer-Verlag, 2001, pp. 90-104.

[34] Jiawei H, Micheline K. Data Mining: Concepts and Techniques. Elsevier, Morgan (2nd Edition) Kaufmann Publishers: San Francisco, 2006.

[35] S.Prabhu and N.Venatesan. Data Mining and Warehousing. New Age International Publishers, Prentice Hall: New Delhi, 2007.

[36] Mihir B ,Ran C, Hugo K, “Keying Hash Functions for Message Authentication”, *Lecture Notes in Computer Science* Vol. 1109 Advances in Cryptology-Crypto 96 N. Springer, Berlin, Heidelberg. 1996, pp 1-19.

[37]Jianming Z,”A New Scheme to Privacy-Preserving Collaborative Data Mining’, *Proc. of Fifth International Conference on Information Assurance and Security,*2009, pp.468-471.

[38] Sin G T, Vincent L, Shuguo H,” A Study of Efficiency and Accuracy of Secure Multiparty Protocol in Privacy-Preserving Data Mining”, *Proc. of 26th International Conference on Advanced Information Networking and Applications Workshops*,2012,pp.85-90.

[39] C. Clifton, “Privately computing a distributed knn classifier,” *Proc. of the Eighth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2004.

[40] Vaidya, J., Yu, H., Jiang, X, Privacy-preserving svm classification. *Knowl. Inf. Syst.* Vol.14, No.2 ,2008, pp. 161–178.

[41] Qingchen Z, Laurence T Y, Zhikui C, and Peng L,” PPHOPCM: Privacy-preserving High-order Possibilistic c-Means Algorithm for Big Data Clustering with Cloud Computing” *IEEE TRANSACTIONS ON BIG DATA*,2016,pp.1-11.

[42] Feng L, Jin M, Jian-hua L,” An Adaptive Privacy Preserving Data Mining Model under Distributed Environment”, *Proc. of Third International IEEE Conference on Signal-Image Technologies and Internet-Based System,IEEE*,2008,pp.60-68.

[43] Ahmed M E, Huaiguo F” Privacy Preserving Distributed Learning Clustering Of HealthCare.Data Using Cryptography Protocols”, *Proc. of 34th Annual IEEE Computer Software and Applications Conference Workshops*,2010,pp.140-145.