Privacy Preserving EM-based Clustering

Luong The Dung
Information Technology Center
VietNam Government Information Security Commission
105 Nguyen Chi Thanh, HaNoi, VietNam
Email: thedung@bcy.gov.vn

Ho Tu Bao
Japan Advanced Institute of Science and Technology
Nomishi, Ishikawa, Japan
Institute of Information Technology, Hanoi, Vietnam
Email: bao@jaist.ac.jp

Abstract—**The problem of privacy-preserving EM-based clustering was solved when the dataset is horizontally partitioned into more than two parts (i.g., more than two computation parties). The aim of this work is to develop a method for the more difficult problem when the dataset is horizontally partitioned into only two parts. The key question is how to compute and reveal only the covariance matrix at various steps of the EM iterative process to the participating parties. We propose a method consisting of several protocols that provide privacy preservation for the computation of covariance matrices and final results without revealing the private information and the means. We also extend the proposed method for a better solution to the problem of privacy preserving k-means clustering.**

I. Introduction

Data mining has emerged as a significant technology for gaining knowledge from vast quantities of data [12]. Data mining allows us to analyze personal data or organizational data, such as customer records, criminal records, medical history, credit records, etc. However, analyzing such data creates threats to privacy and thus, might prevent data mining works. The challenge then is whether we can obtain results of mining while still preserve the data secrecy. Privacy preserving data mining (PPDM) techniques have been proposed to address this type of problem [3], [7], [28].

In general, there are mainly two kinds of privacy preserving data mining methods: the randomization methods and the cryptographic methods. The former methods randomize original data or add noise into original data, so the miner cannot see the original data [1], [8], [9]. In the mining process, the miner has to reconstruct approximate distribution of the original data set from random values [2]. The latter methods fall under the theoretical framework of secure multiparty computation [10]. These techniques allow two or many parties to cooperate for computation works on their joint data sets without disclosing each party's private data to other parties [6], [7], [15], [27]. Cryptographic methods have also been applied to various data mining tasks such as association rules mining [28], classification [16], clustering [15], etc. The problem that we describe in this paper is a special cases of the secure multiparty computation problem.

Clustering is one of the most important techniques of data mining. The task is to group similar objects in a given data set into clusters with the goal of minimizing an objective function [5], [12]. Clustering is widely used in many applications such as customer behaviour analysis, targeted marketing, and others. Recently, privacy preserving clustering problems have also been studied by many authors. In [23] and [24], the authors focused on different transformation techniques that enable the data owner to share the data with the other party who will cluster it. Clifton and Vaidya proposed a secure multi-party computation of k-means algorithm on vertically partitioned data [29]. In [13], the authors proposed a solution for privacy preserving clustering on horizontally partitioned data, where they primarily focused on hierarchical clustering methods that can both discover clusters of arbitrary shapes and deal with different data types. In [15], Kruger et al. proposed a privacy preserving, distributed k-means protocol on horizontally partitioned data that the key step is privacy preserving of cluster means. At each iteration of the algorithm, only means are revealed to parties without other things. But, revealing means might allow the parties to learn some extra information of each other (this problem will be analyzed in section 6).

EM is an important cluster technique. To our knowledge, there is so far only one secure method for the expectation maximization (EM) mixture model from horizontally distributed sources [17]. The basic idea of this method is that in each iteration, each party creates a local model from its data objects and computes global information from the previous iteration, then it securely merges its local model with the other's ones to generate the global model. This provides sufficient information to compute the global information needed for the next iteration. Once this process converges, each party can determine the clusters for its objects. However, this method requires at least three participating parties. Because the global model is a sum of local models, in case only two parties, which often happens in practice, each party could compute other party's local model by subtracting its local model from the global model.

The objective of our work is to develop a privacy preserving EM-based clustering method for horizontally partitioned data on only two parties. This problem requires protecting privacy of intermediate global information in particular the means vector of each cluster, and thus the computation of secure sum should be different from, and more complicated than, that for more than two parties. To allow parties to obtain the final results without revealing intermediate candidate cluster centers, we propose some protocols for secure computation works, such as the covariance matrix, means and the posterior probability computation. We also give the extension of the proposed method to design the algorithm of the privacy preserving k-means clustering for horizontally partitioned data on two parties. Unlike the privacy preserving k-means algorithm developed in [15], our algorithm does not reveal intermediate candidate cluster centers. Thus, parties cannot learn extra information of the others.

The rest of this paper is organized as follows. Next, in section 2, we briefly discuss related background, such as the EM algorithm and the security model. In section 3, we present the privacy preserving EM-based clustering protocol. In section 4, we present the related sub-protocols. In section 5, using the standard method of evaluating the protocols in PPDM such as in [6], [28], etc. we provide an analysis and the estimation method of communication cost to prove (evaluate) the validity of our proposed methods. After that, section 6 gives an extension of the proposed method to address the problem of privacy preserving k-means clustering, and finally, section 7 concludes our work.

II. Background

A. EM algorithm

In this section, we review the EM algorithm over Gaussian Mixture Model, more details on the EM algorithm and mixture models can be found in [5] and [19].

Let D be a data set that has m objects $\{x_1,\ ...,\ x_m\}$ described by n attributes. Denote $x_j = (x_j[1],\ x_j[2],\ ...,\ x_j[n])$ the attribute vector of x_j. Assume that there exist k classes in the data set D, each follows some Gaussian distribution. The parameters of the class i are $\psi_i = \{\mu_i, \Sigma_i, \pi_i\}$, in which $\mu_i = (\mu_i[1],\ ...,\ \mu_i[n])$ is the center of the Gaussian distribution, Σ_i is the covariance matrix of the distribution and π_i is the probability of the class i. The normal density function of class i can be represented by

$$f(x;\psi_i) = \frac{|\Sigma_i^{-1}|^{1/2}}{(2\pi)^{n/2}} exp(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i))$$

Thus, given the unknown parameters set $\psi = \{\psi_1,\ ..,\psi_k\}$, the normal mixture model is

$$f(x;\psi) = \sum_{i=1}^{k} \pi_i f(x;\psi_i)$$

The likelihood of the set data D is represented by

$$L(D;\psi) = \prod_{j=1}^{m} f(x_j;\psi)$$

The maximum likelihood principle says that the estimators that maximize the data likelihood are consistent estimators of the true parameters. The maximizing the likelihood of the set D is usually transformed to an equal maximization problem on the following variable, called log likelihood.

$$log(L(D;\psi)) = -\frac{1}{2}log(2\pi) - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{m} Z_{ij}(log|\Sigma_i| + \frac{1}{2}(x_j-\mu_i)^T\Sigma_i^{-1}(x_j-\mu_i))$$

where Z_{ij} is the posterior probability of x_j from class i, it can be calculated by

$$Z_{ij} = \frac{f(x_j;\psi_i)\pi_i}{\sum_{l=1}^{k} f(x_j;\psi_l)\pi_l} \quad (1)$$

The EM algorithm is to estimate the parameters set ψ. To estimate ψ, it starts with a randomly chosen initial parameter configuration ψ^0. Then, it keeps invoking iterations to recompute ψ^{t+1} based on ψ^t. Every iteration consists of two steps:

E-step: computes the expected value of Z_{ij}.

M-step: updates the parameters ψ^{t+1} using the following equations

$$\mu_i^{(t+1)} = \frac{\sum_{j=1}^{m} Z_{ij}^{(t)} x_j}{\sum_{j=1}^{m} Z_{ij}^{(t)}} \quad (2)$$

$$\Sigma_i^{(t+1)} = \frac{\sum_{j=1}^{m} Z_{ij}^{(t)}(x_j-\mu_i^{(t+1)})(x_j-\mu_i^{(t+1)})}{\sum_{j=1}^{m} Z_{ij}^{(t)}} \quad (3)$$

$$\pi_i^{(t+1)} = \frac{\sum_{j=1}^{m} Z_{ij}^{(t)}}{m} \quad (4)$$

The algorithm stops when $|log(L(\psi^{(t+1)}) - log(L(\psi^{(t)})| < \epsilon$, where ϵ is a preselected threshold.

B. The security model

The privacy preservation of the proposed protocols based on the semi-honest security model. In this model, each party participating in the protocol have to follow the rules using its correct input, and it cannot use what it sees during execution of the protocol to compromise the security. This model is reasonable to many real situations because the parties who want to mine data for their mutual benefit will follow the protocol to get correct results. The definition of secure two party computation in the semi-honest model is stated in [10]. Basically, the definition states that a computation is secure if the view of each party during the execution of the protocol can be effectively simulated by the input and the output of the party.

In this paper, we also use Composition theorem for the semi-honest model that its discussion and the proof can be found in [10].

Composition theorem. *Suppose that g is privately reducible to f and that there exists a protocol for privately computing f. Then there exists a protocol for privately computing g.*

III. Privacy preserving EM-based clustering

Assume that the data set D is horizontally partitioned into two parties Alice and Bob. Alice (party 1) has the data set D_1 including m_1 objects $\{x_1,\ ...,\ x_{m_1}\}$, and Bob (party 2) has the data set D_2 including m_2 objects $\{x_{m_1+1},\ ...,\ x_m\}$, where $m_1 + m_2 = m$. Assume that Alice and Bob want to cluster the joint data set without revealing anythings except for the final results. So, each party could learn the cluster to which each of their data objects belongs, but they learn nothing else. We are assuming that clustering on the joint data set of the two parties is more desirable than clustering on the two data sets individually.

As already pointed out in section 2, the goal of the cluster algorithm is to compute Z_{ij}. To obtain Z_{ij}, each party need to know the covariance matrix Σ_i, the vector of means μ_i and π_i in each iteration of the algorithm. We rewrite the equations for computing these parameters as follows:

$$\begin{aligned}\mu_i^{(t+1)} &= \frac{A_{i1}+A_{i2}}{C_{i1}+C_{i2}}\\ \Sigma_i^{(t+1)} &= \frac{B_{i1}+B_{i2}}{C_{i1}+C_{i2}}\\ \pi_i^{(t+1)} &= \frac{C_{i1}+C_{i2}}{m_1+m_2}\end{aligned}$$

where

$$A_{il} = \sum_{x_j \in D_l} Z_{ijl}^{(t)} x_j \qquad (5)$$

$$B_{il} = \sum_{x_j \in D_l} Z_{ijl}^{(t)} (x_j - \mu_i^{(t+1)})(x_j - \mu_i^{(t+1)}) \qquad (6)$$

$$C_{il} = \sum_{x_j \in D_l} Z_{ijl}^{(t)} \qquad (7)$$

and Z_{ijl} is the posterior probability of x_j from class i, $x_j \in D_l$ ($l = 1$ or 2). Denote $\{Z_{ijl}\}$ be the set of all Z_{ijl} values.

Clearly, A_{il}, B_{il},C_{il} values can be computed locally at each site. Therefore, a way for preserving privacy in clustering is that, we use a protocol to calculate and share μ_i, Σ_i and π_i values without knowing the shared numerator and shared denominator. Thus, to solve this problem, we can use the Division protocol in [7]. This protocol is based on a secure scalar product protocol for 2 parties. After obtaining the global parameters μ_i, Σ_i and π_i. Z_{ijl} can be computed locally by equation (1).

However, we should note that sharing Σ_k to each party does not make privacy breaches [27], but sharing means might allow parties to learn some information of each other (this problem will be explained at the end of this section). Thus, we are proposing a more secure solution, which allows each party to obtain the final result without sharing means. We decompose the privacy preserving EM-based clustering problem into three following sub-problems and propose sub-protocols for them in section 4:

Problem 1. (secure mean computation) Party 1 has a pair (n, x), where x is a real number and n is a positive integer. Similarly, party 2 has a pair (m, y). They want to jointly compute $\frac{x+y}{n+m}$ that party 1 obtains the random value r_1 without other information, party 2 obtains r_2 without other information, where $r_1 + r_2 = \frac{x+y}{n+m}$. In other words, we need to design a secure computation protocol for the following functionality:

$$((n, x), (m, y)) \longmapsto (r_1, r_2) | r_1 + r_2 = \frac{x+y}{n+m}$$

Problem 2. (secure covariance matrix computation) Party 1 (resp. Party 2) has the data set D_1 (resp. D_2) as described above. Party 1 (resp. Party 2) has the μ_{i1} vector and the $\{Z_{ij1}\}$ values set (resp. μ_{i2} and $\{Z_{ij2}\}$), where $\mu_{i1}+\mu_{i2} = \mu_i$, $\{Z_{ijl}\}$ and μ_i are described in the EM algorithm. They want to jointly compute Σ_i as equation (3), thus both parties obtain Σ_i without disclosing the local values. In other words, we need to design a secure computation protocol for the following functionality:

$$((D_1, \mu_{i1}, \{Z_{ij1}\}), (D_2, \mu_{i2}, \{Z_{ij2}\})) \longmapsto (\Sigma_i, \Sigma_i)$$

Problem 3. (secure posterior probability computation) Party 1 (resp. Party 2) has μ_{i1} (resp. μ_{i2}) where $\mu_{i1} + \mu_{i2} = \mu_i$. Both party share π_i and Σ_i, where π_i and Σ_i are the parameters described in the EM algorithm. One party (assume Party 1) having an object x_j wants to compute Z_{ij1} (the posterior probability of x_j from class i), it can cooperate with Party 2 to compute Z_{ij1}, thus Party 1 obtains Z_{ij1}, Party 2 obtains nothing. In other words, we need to design a secure computation protocol for the following functionality:

$$((\Sigma_i, \pi_i, \mu_{i1}, x_j), (\Sigma_i, \pi_i, \mu_{i2})) \longmapsto (Z_{ij1}, \mathit{nothing})$$

Designing protocols for the above problems is presented in next section. We present the privacy preserving EM-based clustering protocol as follows

Protocol 1. Privacy preserving EM-based clustering
Input: Alice and Bob have sets D_1 and D_2, respectively
Output: Each party knows the cluster to which each of their data objects belongs

```
1:  Each party $l$ randomly initialize $z_{ijl}$ to 0 or 1 ( $i = 1...k$,
    $j = 1...m_l$).
2:  t=0
3:  while $\delta < \epsilon$ do
4:      for $i = 1...k$ do
5:          Each party $l$, calculate $A_{il}^{(t+1)}$ and $C_{il}^{(t+1)}$ using
            equations (5) and (7).
6:          Two parties jointly compute $\mu_i^{(t+1)}$ using the secure
            mean computation (Protocol 2). Each party $l$ obtains
            $\mu_{il}^{(t+1)}$ without other things.
7:          Two parties jointly compute $\Sigma_i^{(t+1)}$ using the secure
            covariance matrix computation protocol (Protocol 3).
            Both parties obtain $\Sigma_i^{(t+1)}$ without other things.
8:          Two parties jointly compute $\pi_i^{(t+1)}$ using the Division
            protocol in [7] (or Protocol 2). Both parties obtain
            $\pi_i^{(t+1)}$ without other things.
9:          Each party $l$ cooperates with the other party to
            compute $Z_{ijl}$ using Protocol 4. Party $l$ obtains $Z_{ijl}$,
            the other party obtains nothing.
10:     end for
11:     $t = t + 1$
12:     Two parties jointly compute the log likelihood differ-
        ence $\delta = |log(L(\psi^{(t+1)}) - log(L(\psi^{(t)})|$
13: end while
```

Analysis of privacy. In Protocol 1, only communication occur at steps 6, 7, 8, 9 and 11. At each step, it uses sub-protocols to compute and share the global information without disclosing private information. thus, assume that the used subprotocols are secure, then applying Composition theorem, we can conclude that Protocol 1 is secure. Indeed, we now check whether the revealed values at the above steps can be used to deduce any information on individual data items.

At step 6, each party l only obtains the random values μ_{il}, they do not know any other information including μ_i, so deductions on these values are not possible. We should note that, the disclosing the global means can allow parties to learn a bit extra information of each other. For example, each party can guess the upper bound and lower bound values of an attribute of the other party. Moreover, if the global means and the number of objects of each party are together disclosed, the parties can deduce the local means and the local covariance matrices at each site, thus the probability that a data point belongs to a specified interval can be calculated at each site.

The Σ_i matrix and the π_i value are shared at step 7 and 8, respectively, by themself, they do not reveal private information, because they are distilled from many data values at sites. At step 9, each party l obtains its Z_{ijl} without sharing for the other party. At step 11, the parties compute the log likelihood difference value, and to obtain this value, each party l needs to share its log likelihood to the other party. Clearly, disclosing the local log likelihood does not make privacy breaches, because the log likelihood is distilled from many data items.

IV. Sub-protocols

A. *Secure mean computation protocol*

In this section, we propose a protocol for the secure mean computation problem based on the oblivious polynomial evaluation. The problem of the oblivious polynomial evaluation was first considered in [20]. As with oblivious transfer, this problem involves a sender and a receiver. The senders input is a polynomial Q of degree k over some finite field F and the receivers input is an element $z \in F$ (the degree k of Q is public). The protocol is such that the receiver obtains $Q(z)$ without learning anything else about the polynomial Q, and the sender learns nothing. An efficient solution to this problem was presented in [21].

Protocol 3. Secure mean computation
Input: Assume that two parties Alice and Bob have $(n,\ x)$ and $(m,\ y)$, respectively.
Output: Alice obtains r_1, Bob obtains r_2

1: Alice uniformly chooses an element p from F and defines the linear polynomial $Q_1(z) = pz + pn$
2: Alice and Bob engage in a private evaluation of Q_1, in which Bob obtains $b_1 = Q_1(m) = pm + pn$
3: Bob chooses a random element $q \in F$ and defines the linear polynomial $Q_2(z) = yz - (pm + pn)q$
4: Alice and Bob engage in a private evaluation of Q_2, in which Alice obtains $a_1 = Q_2(p) = py - (pn + pm)q$
5: Alice chooses a random element $r \in F$ and defines the linear polynomial $Q_3(z) = -rz + py + px - (pn + pm)q$
6: Alice and Bob engage in a private evaluation of Q_3, in which Bob obtains $b_2 = Q_3(pn + pm) = -r(pn + pm) + py + px - (pn + pm)q$
7: Alice has $r_1 = r$ and Bob computes $r_2 = \frac{b_2}{b_1} + q = -r + \frac{x+y}{n+m}$ So, the respective outputs of Alice and Bob are r_1 and r_2, giving us that $r_1 + r_2 = \frac{x+y}{n+m}$

Analysis of privacy. We can easily prove that Protocol 4 constitutes a private protocol for computing the mean value as stated. Indeed, we can show that each party's view of the protocol can be simulated based on its input and its output.

During execution of the protocol, Alice only sees the message $a_1 = py - (pn + pm)q$, where q uniformly selected from F, and y, n and m are constants. Assume that $p' \in F(\neq p)$, a_1'=$p'y - (p'n + p'm)q$, we have a_1 and a_1' be the uniform distribution on a specified set, the probability that Alice see some values during the execution is $1/|F|$. Therefore, the two ensembles a_1 and a_1' are statistically indistinguishable. In other word, the simulator for Alice will be a uniform number generator. Similarly, Bob sees the messages $pm + pn$ and $-r(pn + pm) + py + px - (pn + pm)q$, these two messages are independent because p and r are independently chosen by Alice, moreover they have the uniform distribution on a specified set. Therefore they can be simulated by uniform number generators.

B. *The secure covariance matrix computation protocol*

We first present the detailed analysis of computing Σ_i and then we give the protocol.

Recall that the Σ_i matrix has m rows and m columns and each element $\Sigma_i(p, q)$ of Σ_i is computed by formula:

$$\Sigma_i(p,q) \quad = \quad \frac{\sum_{j=1}^{m} Z_{ij}(x_j[p] - \mu_i[p])(x_j[q] - \mu_i[q])}{\sum_{j=1}^{m} Z_{ij}}$$

where $x_j[p]$ and $x_j[q]$ are p^{th} and q^{th} values of the data vector x_j; $\mu_i[p]$ and $\mu_i[q]$ are p^{th} and q^{th} values of the means vector μ_i.

Assume that with each $\mu_i[p](p = 1,\ ...,\ n)$, there exist $\mu_{i1}[p]$ (belong to Alice) and $\mu_{i2}[p]$ (belong to Bob), where $\mu_{i1}[p] + \mu_{i2}[p] = \mu_i[p]$ and existing two values α and β belong to Alice and Bob, respectively, $\alpha + \beta = 1/\sum_{j=1}^{m} Z_{ij}$. We rewrite:

$$\Sigma_i(p,q) = (a_1,\ b_1, c_1,\ d_1,\ e_1,\ f_1, g_1,\ 1) \times \begin{pmatrix} \beta \\ \beta g_2 \\ \beta f_2 \\ \beta e_2 \\ \beta d_2 \\ \beta c_2 \\ \beta b_2 \\ \beta a_2 \end{pmatrix}$$

$$+(\alpha a_1,\ \alpha b_1, \alpha c_1,\ \alpha d_1,\ \alpha e_1,\ \alpha f_1, \alpha g_1,\ \alpha) \times \begin{pmatrix} 1 \\ g_2 \\ f_2 \\ e_2 \\ d_2 \\ c_2 \\ b_2 \\ a_2 \end{pmatrix}$$

where,

$$\begin{aligned}
a_l &= \sum_{x_j \in D_l} Z_{ijl}(x_j[p] - \mu_{il}[p])(x_j[q] - \mu_{il}[q]), \\
b_l &= -\sum_{x_j \in D_l} Z_{ijl}(x_j[p] - \mu_{il}[p]), \\
c_l &= -\sum_{x_j \in D_l} Z_{ijl}(x_j[q] - \mu_{il}[q]), \\
d_l &= Z_{ijl}, \\
e_l &= \mu_{il}[p]\mu_{il}[q], \\
f_l &= \mu_{il}[p], \\
g_l &= \mu_{il}[q]
\end{aligned}$$

Denote:

$$\begin{aligned}
U_1 &= (a_1,\ b_1,\ c_1,\ d_1,\ e_1,\ f_1, g_1,\ 1) \\
V_1 &= (\ \beta,\ \beta g_2,\ \beta f_2,\ \beta e_2,\ \beta d_2,\ \beta c_2,\ \beta b_2,\ \beta a_2)^T \\
U_2 &= (\alpha a_1,\ \alpha b_1,\ \alpha c_1,\ \alpha d_1,\ \alpha e_1,\ \alpha f_1, \alpha g_1,\ \alpha) \\
V_2 &= (1,\ g_2,\ f_2,\ e_2,\ d_2,\ c_2,\ b_2,\ a_2)^T
\end{aligned}$$

We have $\Sigma_i(p,q) = U_1 \bullet V_1 + U_2 \bullet V_2$

It should be noted that U_1 and U_2 can be computed by Alice alone; V_1 and V_2 can be computed by Bob alone. Therefore, Alice and Bob can compute the dot products $U_1 \bullet V_1$ and $U_2 \bullet V_2$ using the scalar product protocol in [11]. Computing Σ_i can be summarized in the following protocol

Protocol 3. Secure covariance matrix computation

Input: Alice has D_1, μ_{i1} and $\{Z_{ij1}\}$, Bob has D_2, μ_{i2} and $\{Z_{ij2}\}$.

Output: Σ_i

```
Alice computes $C_{i1}$ and Bob computes $C_{i2}$ using equation (7). Alice and Bob jointly compute $1/(C_{i1} + C_{i2})$ using Protocol 2. Alice and Bob obtain $\alpha$ and $\beta$, respectively, where $\alpha + \beta = 1/(C_{i1} + C_{i2})$.
for all the pairs ($p$, $q$) do
    Alice presents the vector $U_1$, Bob presents the vector $V_1$. Two parties jointly compute the dot product of $U_1$ and $V_1$. Alice obtains $P_1$ and Bob obtains $Q_1$, where $P_1 + Q_1 = U_1 \bullet V_1$.
    Alice presents the vector $U_2$, Bob presents the vector $V_2$. Two parties jointly compute the dot product of $U_2$ and $V_2$. Alice obtains $P_2$ and Bob obtains $Q_2$, where $P_2 + Q_2 = U_2 \bullet V_2$.
    Alice sends $P_1+P_2$ to Bob. Bob sends $Q_1+Q_2$ to Alice, both parties obtain $\Sigma_i(p,q)$ without revealing private information.
end for
```

C. Secure posterior probability computation protocol

Assume the computation of Σ_i is implemented at the iteration t^{th} of the algorithm, then at the iteration $(t+1)^{th}$ if one party (assume Alice) having an object $x_j = (x_j[1]$, ..., $x_j[n])$, it needs to determine Z_{ij} for x_j.

To obtain Z_{ij}, Alice needs to obtain $T_j = (x_j - \mu_i)^T \Sigma_i^{-1}(x_j - \mu_i)$ and then he can compute Z_{ij} using equation (1).

We have

$$\begin{aligned}
&(x_j - \mu_i) = (x_j[1] - \mu_i[1],\ ...,\ x_j[n] - \mu_i[n]) \\
&= ((x_j[1] - \mu_{i1}[1]) - \mu_{i2}[1],\ ...,\ (x_j[n] - \mu_{i1}[n]) - \mu_{i2}[n])
\end{aligned}$$

Denote

$$\begin{aligned}
\alpha_1[q] &= \sum_{p=1}^{n} (x_j[p] - \mu_{i1}[p])\Sigma_i(p,q) \\
\beta_1[q] &= -\sum_{p=1}^{n} \mu_{i2}[p]\Sigma_i(p,q) \\
\alpha_2[q] &= x_j[q] - \mu_{i1}[q]) \\
\beta_2[q] &= -\mu_{i2}[q]
\end{aligned}$$

We rewrite the equation of T_j

$$
\begin{aligned}
T_j &= \sum_{q=1}^{n}(\alpha_1[q]\alpha_2[q] + \alpha_1[q]\beta_2[q] + \alpha_2[q]\beta_1[q] + \beta_1[q]\beta_2[q]) \\
&= (\sum_{q=1}^{n}\alpha_1[q]\alpha_2[q],\ \alpha_1[1],\ \ldots\ \alpha_1[n]) \begin{pmatrix} 1 \\ \beta_2[1] \\ . \\ . \\ . \\ \beta_2[n] \end{pmatrix} \\
&+ (\alpha_2[1],\ \ldots\ \alpha_2[n], 1) \begin{pmatrix} \beta_1[1] \\ . \\ . \\ . \\ \beta_1[n] \\ \sum_{q=1}^{n}\beta_1[q]\beta_2[q] \end{pmatrix} \\
&= U_1 \bullet V_1 + U_2 \bullet V_2
\end{aligned}
$$

where, U_1 and U_2 can be computed by Alice alone; V_1 and V_2 can be computed by Bob alone. Therefore Alice and Bob can compute the dot products $U_1 \bullet V_1$ and $U_2 \bullet V_2$ using the scalar product protocol in [11]. Thus Alice can obtain T_j without disclosing private information and then he can compute Z_{ij}. Computing Z_{ij} can be summarized in the following protocol.

Protocol 4. Secure posterior probability computation
Input: Alice has D_1, x_j and μ_{i1}, Bob has D_2 and μ_{i2}, both know Σ_i and π_i.
Output: Alice obtains Z_{ij}, Bob obtains nothing.

1: Alice presents the vector U_1, Bob presents V_1
2: Alice and Bob jointly compute the dot product $U_1 \bullet V_1$. Alice obtains X_1, Bob obtains Y_1.
3: Alice presents the vector U_2, Bob presents V_2
4: Alice and Bob jointly compute the dot product $U_2 \bullet V_2$. Alice obtains X_2, Bob obtains Y_2.
5: Bob sends $X_2 + Y_2$ to Alice
6: Alice obtains $T_j = X_1 + Y_1 + X_2 + Y_2$. Bob obtains nothing.
7: Alice computes Z_{ij} using equation (1)

We should note that Protocol 3 and Protocol 4 use the secure scalar product protocol and the secure mean computation protocol (Protocol 2), where there already exist many scalar product protocols that are correct and secure [7], [28], [11]. During the execution of this protocol, the parties participating in the protocol are not able to learn anything other than the final result. So, applying Composition theorem, we can conclude that Protocol 3 and Protocol 4 are secure.

V. The Communication Analysis

We give an analysis of the communication cost of the privacy preserving EM-based clustering protocol at its one iteration. The total cost is dependent on the number of iterations required to converge, which is dependent on the data. Assume that the communication cost of Protocol 1, Protocol 2, Protocol 3 and Protocol 4 are P_1, P_2, P_3 and P_4, respectively. We should note that the communication cost of the scalar product protocol is $O(n)$ (see in [11]), where n is the size of input vectors and the communication cost of the oblivious polynomial evaluation protocol is $O(k)$ exponentiations [20] or $O(k|F|)$ (see in [21]), where k is the degree of the input polynomial and $|F|$ is the size of the field used and depends on the range of the variables in calculating. In one iteration of the Protocol 1, the communication occurs at steps 6, 7, 8 , 9 and 11. Then, its the communication cost is $P_1 = (n+1)P_2 + P_3 + P_4$.

Protocol 2 calls the oblivious polynomial evaluation protocol three times (with the degree 1 polynomials), so $P_2 = O(1)$. Protocol 3 calls Protocol 2 one time and calls the scalar product protocol $2n^2$ times with the size of input vectors 8, $P_3 = O(n^2)$. Similarly, $P_4 = O(1)$. Finally, we have $P_1 = (n+1)O(1) + O(n^2) + O(1) = O(n^2)$. In fact, n is small, thus this is quite reasonable.

VI. Extend to the k-means algorithm

In an iteration of the k-means algorithm, to determine the cluster of an object, the Euclidian distances between the object and the cluster centers need to be computed. Thus, to preserve privacy for k-means clustering, the key step is privacy preserving of the cluster means. At each iteration of the algorithm, only means are revealed to parties without other things [15]. However, the problem is that revealing means might make privacy breaches, for example, each party can guess the upper bound and lower bound values of an attribute of other party. Moreover, at early iteration steps of the clustering algorithm, means are computed from only one or a few original values of parties, thus each party might guess original values of the other party based on means. In order to overcome this problem, we design a protocol, which allows each participating party to compute the distances between its objects and the clusters center without revealing the clusters center.

Assume that at an iteration of the algorithm, Alice has C_k^A and vector $x = (x[1],\ \ldots,\ x[n])$, Bob has C_k^B, where $C_k = C_k^A \cup C_k^B$ is a cluster. Alice needs to compute the distance (d) between x and the center of cluster k. The distance d is computed by the formula:

$$d = (x[1] - \mu_k[1])^2 + \ldots + (x[n] - \mu_k[n])^2$$

Assume that $\mu_k[i] = a[i] + b[i]$, we rewrite

$$
\begin{aligned}
d &= (x[1] - a[1] - b[1])^2 + \ldots + (x[n] - a[n] - b[n])^2 \\
&= \sum_{i=1}^{n}(x[i] - a[i])^2 - 2\sum_{i=1}^{n}(x[i] - a[i])b[i] + \sum_{k=1}^{n}(b[i])^2
\end{aligned}
$$

We note that

$$\sum_{k=1}^{n}(x[i]-a[i])b[i] =$$

$$(x[1]-a[1],\ ...,\ x[n]-a[n]) \times \begin{pmatrix} b[1] \\ \\ . \\ . \\ . \\ b[n] \end{pmatrix} = R^A \bullet R^B$$

Therefore, d can be computed by the scalar product protocol. The privacy preserving Euclidian distance protocol can be summarized as follows.

1: Alice and Bob jointly compute the means vector μ_k using Protocol 2. Alice obtains $a = (a[1],\ ...,\ a[n])$, Bob obtains $b = (b[1],\ ...,\ b[n])$

2: Alice presents vector R^A, Bob presents vector R^B.

3: Alice and Bob jointly compute the dot product of R^A and R^B using the scalar product protocol. Alice obtains X^A, Bob obtains X^B.

4: Bob computes $Y^B = -2X^B + \sum_{k=1}^{n}(b[i])^2$ and sends it to Alice

5: Alice computes $d = \sum_{i=1}^{n}(x[i]-a[i])^2 - 2X^A + Y^B$

VII. Conclusion

In this paper, we have developed the privacy preserving EM-based clustering method for horizontally partitioned data on two parties. We have presented some protocols based on oblivious polynomial evaluation and the secure scalar product for addressing some problems, such as the means, covariance matrix and posterior probability computation. We have also given an extension of the proposed method to address the problem of privacy preserving k-means clustering. Our method allows two parties to cooperatively conduct clustering on their joint data sets without disclosing each party's private data to the other.

References

[1] S.Agrawal and JR.Haritsa (2005). A Framework for High-Accuracy Privacy-Preserving Mining. In Proceedings 21st International Conference on Data Engineering, ICDE 2005, Japan.

[2] R.Agrawal, R.Srikant, and D.Thomas. Privacy preserving OLAP. In Proceedings of the 2005 ACM SIGMOD international Conference on Management of Data (Baltimore, Maryland, June 14 - 16, 2005). SIGMOD 05. ACM, New York, NY, 251- 262.

[3] Y.Chang and M.Mitzenmacher. Privacy Preserving Keyword Searches on Remote Encrypted Data. Cryptology ePrint Archive: Report 2004/051 J. Vaidya and C. Clifton.

[4] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM05), pages 589592, Houston, TX, November 2005

[5] A. P. Dempster; N. M. Laird; D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). J Roy Stat Soc B 39:138.

[6] W.Du, Yunghsiang S. Han and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. Proceedings of the Fourth SIAM

[7] W.Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In Proc. of the 17th Annual Computer Security Applications Confer- ence, pages 103110, 200

[8] A.Evfimievski. Randomization in Privacy Preserving Data Mining. ACM SIGKDD Explorations Newsletter, Volume 4, Issue 2, Pages: 43-48, December 2002.

[9] A. Evfimevski, J. Gehrke, and R. Srikant. Limiting Privacy Breaches in Privacy Preserving Data Mining. Proc. ACM SIGMOD/PODS Conf., June 2003.

[10] O. Goldreich. Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, 2004.

[11] B.Goethals, S.Laur, H.Lipmaa, and T.Mielikainen. On private scalar product computation for privacy-preserving data mining. In Proc. of the Seventh Annual International Conference in Information Security and Cryptol- ogy, LNCS. Springer-Verlag, 2004. to appear.

[12] J. Han and M. Kamber. Data Mining Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

[13] A.Inan, Y.Saygn, E.Sava, A.Azgn Hintolu, A.Levi. Privacy Preserving Clustering on Horizontally Partitioned Data. Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)

[14] I. Ioannidis, A. Grama, and M. Atallah. A secure protocol for computing dot-products in clustered and distributed environments. In The 2002 International Conference on Parallel Processing, 2002.

[15] S.Jha, L.Kruger, McDaniel. Privacy Preserving Clustering. In Proc. of the 10th European Symposium on Research in Computer Security, p.397-417, 2005.

[16] M. Kantarcoglu and J. Vaidya. Privacy Preserving Naive Bayes Classifier for Horizontally Partitioned Data. In IEEE ICDM Workshop on Privacy Preserving Data Mining, pp. 3-9, 2003.

[17] X.Lin, C.Clifton, and M.Zhu, 2005. Privacy-preserving clustering with distributed EM mixture modeling. Knowl. Inf. Syst. 8, 1 (Jul. 2005), 68-81.

[18] K.Liu, H.Kargupta, and J.Ryan. Random Projection-based Multiplicative Perturbation for Privacy Preserving Distributed Data Mining. IEEE Transactions on Knowledge and Data Engineering (TKDE), VOL. 18, NO. 1, pages 92–106, Piscataway, NJ, January 2006.

[19] G.J.McLachlan, K.E.Basford (1988) Mixture models: inference and applications to clustering. Dekker, New York

[20] M. Naor and B. Pinkas, Oblivious Transfer and Polynomial Evaluation, Proceedings of the 31th Annual Symposium on the Theory of Computing (STOC), ACM, 1999, pp. 245254.

[21] M. Naor and B. Pinkas. Efficient Oblivious Transfer Protocols, Proceedings of 12th SIAM Symposium on Discrete Algorithms (SODA), January 7-9 2001, Washington DC, pp. 448457.

[22] B. Pinkas. Cryptographic Techniques for Privacy Preserving Data Mining. SIGKDD Explorations, vol. 4, no. 2, pp. 12-19, 2002

[23] S.R. M. Oliveira, O. R.Zaane. Privacy Preserving Clustering By Data Transformation, In Proc. of the 18th Brazilian Symposium on Databases, p.304-318, 2003.

[24] S.R. M. Oliveira, O. R.Zaane. Achieving Privacy Preservation When Sharing Data for Clustering. In Proc. of the International Workshop on Secure Data Management in a Connected World, p.67-82, 2004.

[25] L.Sweeney. k-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Volume 10, Issue 5, pp. 557-570, 2002.

[26] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), pp. 571-588, 2002.

[27] Luong, T.D., Ho, T.B. Privacy Preserving for Multivariate Outlier Detection, Third International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2008), December 22-23, Hanoi, 7-16.

[28] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 639644. ACM Press, 2002.

[29] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 206215, 2003.

[30] Y.Zhu, and L.Liu. Optimal randomization for privacy preserving data mining. In Proceedings of the Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Seattle, WA, USA, August 22 - 25, 2004).