EM-Based Clustering with Privacy Preserving

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 aim of this work is to propose a privacy-preserving EM-based clustering algorithm for horizontally partitioned data sets between two parties. To this end, we propose basic protocols based on oblivious polynomial evaluation and prove the secrecy of protocols based on the semi-honest security model and the composition theorem. We have also given an extension of the proposed method to address 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 [1]. Data mining technology allows us to analyze personal data or organizational data, such as customer records, criminal records, medical history, credit records, etc. However, analyzing such data create 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 techniques have been proposed to address this type of problem [6], [7], [25], [33], [34].

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 [18], [19], [26]. In the mining process, the miner has to reconstruct approximate distribution of the original data set from random values [21], [22]. The latter methods fall under the theoretical framework of secure multiparty computation [16], [17]. 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], [5], [13], [15]. Cryptographic methods have also been applied to various data mining work such as association rules mining [7], classification [14], clustering [15], etc. The problem that we describe in this paper are 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 [1], [2]. Clustering is widely used in many applications such as customer behaviour analysis, targeted marketing, and others. Privacy preserving clustering problems have also been studied by various authors. In [36] and [37], the authors focused on different transformation techniques that enable the data owner to share the data with another party who will cluster it. Clifton and Vaidya proposed the secure multi-party computation of k-means algorithm on vertically partitioned data in [35]. In [38], the authors proposed the 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 [4], 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 reveal to parties without other things. But, revealing means might allow parties to learn some extra information of each other (this problem will be analyzed in section 4).

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 [3]. The basic idea of this method is that in each iteration, each party creates a local model from its data points and global information from the previous iteration, each party securely merge its model with others 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 cluster for its objects. The limitation of this method is that it requires at least three participating parties. Because the global model is a sum of local models, in case with only two parties, each party could compute other party's local model by subtracting its the local model from the global model.

The objective of our work here is to develop a privacy preserving algorithm for EM-based clustering for horizontally partitioned data on two parties. Moreover, the proposed method can be extended 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 [4], our algorithm does not reveal intermediate candidate cluster centers. Thus, parties can not learn extra information of the others.

The problem presented in this paper also related to the work of Du and Han [5]. Du and Han presented privacy preserving multivariate classification protocols based on the matrix transformation technique, where the core is to compute the covariance matrix with privacy preservation. However, this work is to solve for the vertically partitioned data situation on two party. As pointed out in this paper, we considers

privacy preserving clustering in horizontally partitioned data sets. Moreover, Du and Han's method is based on a heuristic security model that it has never been proved. Our method is based on the cryptographic techniques and the security model with sound proofs..

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. Section 3, we present the privacy preserving EM-based clustering algorithm in horizontally model on two parties and related protocols. Section 4, using the standard method of evaluating the protocols in PPDM such as in [5], [6], [27], etc. we provide an analysis and the estimation method of communication cost to prove (evaluate) the validity of our proposed methods. After that, section 5 gives an extension of the proposed method to address the problem of privacy preserving k-means clustering, and finally, section 6 concludes our work.

II. Background

A. EM algorithm

In this paper we consider EM-based clustering technique, that a brief overview of this technique is presented in [2].

Let D be a data set that has m objects $\{x_1, ..., x_m\}$ described by n attributes. We denote $x_i = (x_i[1], x_i[2], ..., x_i[n])$ the attribute vector of x_i. Assume that there exist C classes in the data set D, each follows some Gaussian distribution. The parameters of the class k include $\{\mu_k, \Sigma_k, \pi_k\}$, in which $\mu_k = (\mu_k[1], ..., \mu_k[n])$ is the center of the Gaussian distribution, Σ_k is the covariance matrix of the distribution and π_k is the probability of the class k. The log likelihood function to be maximized is then given by:

$$L(\mu_k, \Sigma_k, \pi_k | x_i) = \sum_{k=1}^{C} \sum_{x_i \in C_k} (log\pi_k - \frac{d}{2} log 2\pi - \frac{1}{2} log|\Sigma_k| - \frac{1}{2}(x_i - \mu_i)^T \Sigma_k^{-1}(x_i - \mu_i))$$

where C_k is the set of points belonging to the class k. Denote

$$T_k = \hat{x_i}^T \Sigma_k^{-1} \hat{x_i} + ln|\Sigma_k|$$

where $\hat{x_i} = (x_i[1] - \mu_k[1], ..., x_i[n] - \mu_k[n])^T$, $\hat{x_i}^T$ is the transpose of $\hat{x_i}$ and Σ_k is the covariance matrix of C_k. Each element of Σ_k is computed by the formula:

$$\Sigma_k(i,j) = \frac{1}{n_k} \sum_{x_p \in D_k} (x_p[i] - \mu_k[i])(x_p[j] - \mu_k[j])$$

The EM algorithm use an iterative procedure where the E-step classifies the data points according to:

$$x_i \in C_k \iff k = argmin_j\ (T_j)$$

and the M-step updates the parameters μ_k and Σ_k.

The EM-based clustering algorithm is summarized as follows

The EM algorithm ([2], section 2.1)

1: initialize all μ_k randomly
2: initialize all Σ_k to be unit matrices
3: set $t = 1$
4: **while** $\exists n : C_n(t) \neq C_n(t-1)$ **do**
5: **for** all x_i **do**
6: compute T_j ($j = 1, .., C$) and find $k = argmin_j\ T_j$
7: assign x_i to the cluster C_k
8: **end for**
9: update μ_k and Σ_k
10: $t = t + 1$
11: **end while**

A well known and often used simplification of this technique is to assume spherically shaped clusters, which reduces the covariance matrices to unit matrices and leads to the well known k-means algorithm.

B. The security model

The privacy preservation of the proposed protocols based 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 can not use what it sees during execution of the protocol to compromise the security. This model is reasonable to many real situation 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 [16]. 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.

Definition 1 Let x and y be inputs of the two parties and $f_1(x,y), f_2(x,y)$ be the desired functionality, i.e., the first party wants to compute $f_1(x,y)$ and the second wants to compute $f_2(x,y)$. Let Π be a two-party protocol to compute f. The view of the first party after having participated in protocol Π (denoted by $VIEW_1^{\Pi}(x,y)$ is $(x, r, m_1, ..., m_t)$, where r are the random bits generated by party 1 and $m_1, ..., m_t$ is the sequence of messages received by party 1, while participating in protocol Π. The view $VIEW_2^{\Pi}(x,y)$ for the second party is defined in an analogous manner. We say that Π privately computes f if there exists probabilistic polynomial-time algorithms, denoted by S_1 and S_2 such that

$$\{S_1(x, f_1(x,y))\}_{x,y} \stackrel{c}{\equiv} \{VIEW_1^{\Pi}(x,y)\}_{x,y}$$

$$\{S_2(x, f_2(x,y))\}_{x,y} \stackrel{c}{\equiv} \{VIEW_2^{\Pi}(x,y)\}_{x,y}$$

In the equation given above, $\stackrel{c}{\equiv}$ denotes statistically indistinguishable. Detailed definitions of these concepts can be found in [16]. In this paper, we also use the Composition Theorem for the semi-honest model that its discussion and the proof can be found in [16].

Theorem 1: (The 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^A including objects $\{x_1, ..., x_{m_1}\}$, and Bob (party 2) has the data set D^B including objects $\{x_{m_1+1}, ..., x_m\}$. We denote C_k^A and C_k^B the set objects belonging to class k for Alice and Bob, respectively. Assume that Alice and Bob want to cluster the joint data set without revealing any things except for the final cluster centers. 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 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, for clustering the goal of the algorithm is to compute T_k. To obtain T_k, each party need to know the covariance matrix Σ_k and the vector of means μ_k in each iteration step of the algorithm. Therefor, a way for preserving privacy in clustering is that, we build privacy preserving protocols for the covariance matrix and means computation. However, we should note that disclosing Σ_k to each party does not make privacy breaches [13], but disclosing means might allow parties to learn some information of each other. 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 other parties based on means. To overcome the limitation of this way, we need to have protocols that allow parties to obtain T_k and Σ_k without revealing any other information. we assume that there has existed Protocol 2 and Protocol 3 for the secure computation of T_k and Σ_k, respectively (Designing these protocols is presented in the next section). We present the privacy preserving EM-based clustering algorithm as follows.

Protocol 1 Privacy preserving EM-based clustering
Input: Alice and Bob have sets D^A and D^B, respectively
Output: Each party knows the cluster of its objects

1: Alice randomly initializes all μ_k and send them to Bob
2: Both parties initialize all Σ_k to be unit matrices
3: **for** all $x_i \in D^A$ **do**
4: Alice computes all T_j and select $k = argmin_j\ T_j$
5: Alice assigns x_i to the cluster C_k
6: **end for**
7: **for** all $x_i \in D^B$ **do**
8: Bob computes all T_j and select $k = argmin_j\ T_j$
9: Bob assigns x_i to the cluster C_k
10: **end for**
11: set $t = 1$
12: **while** $\exists n : C_n(t) \neq C_n(t-1)$ **do**
13: **for** all $x_i \in D^A$ **do**
14: Two parties jointly compute all T_j using Protocol 2. Alice obtains all T_j. Bob obtains nothing.
15: Alice selects $k = argmin_j\ T_j$ and assign x_i to C_k
16: **end for**
17: **for** all $x_i \in D^B$ **do**
18: Two parties jointly compute all T_j using Protocol 3. Bob obtains all T_j. Alice obtains nothing.
19: Bob selects $k = argmin_j\ T_j$ and assign x_i to C_k
20: **end for**
21: Two parties jointly compute all Σ_k using Protocol 2. Both parties obtain all Σ_k
22: $t = t + 1$
23: **end while**

A. The secure computation protocol for T_k and Σ_k

In order, to create a privacy-preserving version of Σ_k and T_k. we first need to address two problems that can be formally defined as follows:

Problem 1 (The 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 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 (The inverse of sum computation) party 1 has a positive integer n. Party 2 has a positive integer m. They want to jointly compute $\frac{1}{n+m}$ that party 1 and party 2 obtain the random values r_1 and r_2, respectively without other information, where $r_1 + r_2 = \frac{1}{n+m}$. In other words, we need to design a secure computation protocol for the following functionality:

$$(n, m) \longmapsto (r_1, r_2) | r_1 + r_2 = \frac{1}{n+m}$$

The Protocol 4 and Protocol 5 for Problem 1 and Problem 2, respectively, were presented in the next section. Using these protocols, we can build protocols to securely compute Σ_k and T_k as follows.

1) Secure computation protocol for Σ_k: The first, we present the detailed analysis of computing Σ_k and then we give the protocol.

We recall that, each element of Σ_k is computed by formula:

$$\Sigma_k(i,j) = \frac{1}{n_k}\Big(\sum_{x_p \in C_k^A} (x_p[i] - \mu[i])(x_p[j] - \mu[j]) + \sum_{x_p \in C_k^B} (x_p[i] - \mu[i])(x_p[j] - \mu[j])\Big) \quad (1)$$

Assume that with each $\mu[i](i = 1, ..., n)$, there exist two values $\mu^A[i]$ (belong Alice) and $\mu^B[i]$ (belong Bob), where

$\mu^A[i] + \mu^B[i] = \mu[i]$. And existing two values r_1^A and r_1^B belong to Alice and Bob, respectively, $r_1^A + r_1^B = \frac{1}{n_k}$. We rewrite:

$$\Sigma_k(i,j) = (r_1^A + r_1^B) \times$$

$$(U_1^A,\ U_2^A, -U_3^A,\ -\mu^A[j],\ -\mu^A[i],\ 1) \times \begin{pmatrix} 1 \\ \mu^B[j] \\ \mu^B[i] \\ U_3^B \\ U_2^B \\ U_1^B \end{pmatrix}$$

Where,

$$U_1^A = \sum_{x_p \in C_k^A} (x_p[i] - \mu^A[i])(x_p[j] - \mu^A[j]) + \mu^A[i]\mu^A[j],$$
$$U_2^A = -\sum_{x_p \in C_k^A} (x_p[i] - \mu^A[i]),$$
$$U_3^A = -\sum_{x_p \in C_k^A} (x_p[j] - \mu^A[j]),$$
$$U_1^B = \sum_{x_p \in C_k^B} (x_p[i] - \mu^B[i])(x_p[j] - \mu^B[j]) + \mu^B[i]\mu^B[j],$$
$$U_2^B = -\sum_{x_p \in C_k^B} (x_p[i] - \mu^B[i]),$$
$$U_3^B = -\sum_{x_p \in C_k^B} (x_p[j] - \mu^B[j])$$

Denote:

$$V^A = (U_1^A,\ U_2^A,\ -U_3^A,\ -\mu[j]^A,\ -\mu[j]^A,\ 1)$$
$$V^B = (1,\ \mu[j]^B,\ \mu[i]^B,\ U_3^B, U_2^B,\ U_1^B)^T$$

It should be noted that V^A can be computed by Alice alone and V^B can be computed by Bob alone. Therefore Alice and Bob can compute the dot product $V^A \bullet V^B$ using the scalar product protocol in [9], where Alice obtains r_2^A without other things and Bob obtains r_2^B without other things, where $r_2^A + r_2^B = V^A \bullet V^B$.

The next, we rewrite:

$$\Sigma_k(i,j) = (r_1^A + r_1^B)(r_2^A + r_2^B)$$
$$= (r_1^A r_2^A,\ r_1^A,\ r_2^A,\ 1) \times \begin{pmatrix} 1 \\ r_2^B \\ r_1^B \\ r_1^B r_2^B \end{pmatrix} = R^A \bullet R^B$$

Where R^A and R^B are vectors $(r_1^A r_2^A,\ r_1^A,\ r_2^A,\ 1)$ and $(1,\ r_2^B,\ r_1^B,\ r_1^B r_2^B)^T$ respectively. Thus, using the scalar protocol allow Alice and Bob to obtain $\Sigma_k(i,j)$ without revealing private information.

Therefor, computing Σ_k can be summarized in the following protocol

Protocol 2 Secure computation for Σ_k
Input: Alice has C_k^A, Bob has C_k^B.
Output: Σ_k

1: Alice computes n values $a_1, ..., a_n$ and the value a, where $a_j = \sum_{x_i \in C_k^A} x_i[j]$ and $a = |C_k^A|$. Analogously, Bob computes n values $b_1,\ ...,\ b_n$ and the value b , where $b_j = \sum_{x_i \in C_k^B} x_i[j]$ and $b = |C_k^B|$.
2: Alice and Bob jointly compute each $\frac{a_j + b_j}{a+b}$ using the Protocol 4. Alice obtains $\mu^A[1], ..., \mu^A[n]$ and Bob obtains $\mu^B[1], ..., \mu^B[n]$, where $\mu^A[j] + \mu^B[j] = \frac{a_j + b_j}{a+b}$.
3: The next, Alice and Bob jointly compute $\frac{1}{a+b}$ using Protocol 5. Alice and Bob obtain r_1^A and r_1^B, respectively, where $r_1^A + r_1^B = \frac{1}{a+b}$.
4: **for** all the pair (i, j) **do**
5: Alice presents the vector V^A, Bob present the vector V^B
6: Alice and Bob jointly compute the dot product of V^A and V^B. Alice obtains r_2^A and Bob obtains r_2^B, where $r_2^A + r_2^B = V^A \bullet V^B$.
7: Alice presents the vector R^A, Bob presents the vector R^B
8: Alice and Bob jointly compute the dot product of R^A and R^B, both parties obtain $\Sigma_k(i,j)$ without revealing private information.
9: **end for**

2) Secure computation protocol for T_k: Assume the computation of Σ_k has implemented at the iteration t^{th} of the algorithm, then at the iteration $(t+1)^{th}$ if one party (assume Alice) has a object $x = (x[1],\ ...,\ x[n])$, want to predict the cluster of x. Alice can cooperate with Bob to compute T_k as follows.

Denote,

$$\hat{x} = (x[1] - \mu[1],\ ...,\ x[n] - \mu[n])$$
$$= ((x[1] - \mu[1]^A) - \mu[1]^B,\ ...,\ (x[n] - \mu[n]^A) - \mu[n]^B)$$

We have

$$T_k = \hat{x}\Sigma_k^{-1}\hat{x}^T + ln|\Sigma_k|\ (k = 1,\ ...,\ c)$$

Because both parties know Σ_k, we can write $\hat{x}\Sigma_k^{-1}\hat{x}^T$ as the dot product of two vectors W^A and W^B. Where, computing each value in W^A is only requires the value of variables $x[i] - \mu^A[i]$ and $\Sigma_k(i,j)$, so it can be calculated by Alice alone. And computing each value in W^B is only requires the value of variables $\mu^B[i]$ and $\Sigma_k(i,j)$, so it can be calculated by Bob alone. Therefor, using the scalar product protocol, Alice

obtains X^A without other things and Bob obtains X^B without other things, where $X^A + X^B = W^A \bullet W^B$.

Computing Σ_k can be summarized in the following protocol

Protocol 3 Secure computation for T_k
Input: Alice has x, μ^A and Σ_k. Bob has μ^B and Σ_k
Output: Alice obtains T_k. Bob obtains nothing

1: Alice presents vector W^A, Bob presents W^B
2: Alice and Bob jointly compute the dot product of W^A and W^B. Alice obtains X^A, Bob obtains X^B.
3: Bob sends X^B to Alice
4: Alice computes $T_k = X^A + X^B + ln|\Sigma_k|$

3) Analysis of Privacy:: We should note that Protocol 1 and Protocol 2 use the secure scalar product protocol and the basic protocols Protocol 3 and Protocol 4, where there already exist many scalar product protocols that are correct and secure [6], [7], [9]. During the execution of this protocols, parties participating in protocols are not able to learn anything other than the final result. So, applying the composition theorem, Protocol 2 and Protocol 3 will be secure if the basic protocols are secure. In next section, we introduce basic protocols, and prove that they are secure.

B. Basic Protocols

In this section, we propose protocols for Problem 1 and Problem 2 based on the oblivious polynomial evaluation. The problem of the oblivious polynomial evaluation was first considered in [10]. 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 [11].

Protocol 4 The mean value computation
Input: 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 $Q_2(z) = \frac{1}{pm+pn}z - q$
4: Alice and Bob engage in a private evaluation of Q_2, in which Alice obtains $a_1 = Q_2(px) = \frac{x}{n+m} - q$.
5: Alice chooses a random element $r \in F$ and defines the linear polynomial $Q_3(z) = pz - r$.
6: Alice and Bob engage in a private evaluation of Q_3, in which Bob obtains $b_2 = Q_3(\frac{y}{pn+pm}) = \frac{y}{n+m} - r$
7: Alice computes $r_1 = a_1 + r = \frac{x}{n+m} - q + r$, Bob computes $r_2 = b_2 + q = \frac{y}{n+m} - r + q$. So, the respective outputs of Alice and Bob are r_1 and r_2, giving us that $r_1 + r_2 = \frac{x+y}{n+m}$.

Proof of privacy: We are proving that, Protocol 4 constitutes a private protocol for computing the mean value as stated in Problem 1. The views of the two parties are

$$VIEW_1(n,\ x) = (n,\ x,\ p,\ r,\ \frac{x}{n+m} - q)$$
$$VIEW_2(m,\ y) = (m,\ y,\ pm + pn, q,\ \frac{y}{n+m} - r)$$

Because the elements p, r are uniformly chosen from F (by Alice) and the element q is uniformly chosen from F (by Bob), we can rewrite the views of the two parties as follows.

$$VIEW_1(n,\ x) = (n,\ x,\ \frac{x}{n+m} - q)$$
$$VIEW_2(m,\ y) = (m,\ y,\ pm + pn,\ \frac{y}{n+m} - r)$$

Let q' and r' be two random elements $\in F$. Let S_1 and S_2 be probabilistic polynomial-time algorithms that they are defined as follows.

$$S_1(n,\ x,\ \frac{x}{n+m} - q) = (n,\ x,\ \frac{x}{n+m} - q + q')$$

$$S_2(m,\ y,\ \frac{y}{n+m} - r) = (m,\ y,\ \frac{r'}{y - r(n+m)}n + \frac{r'}{y - r(n+m)}m,\ \frac{y}{n+m} - r + r')$$

It is straightforward that S_1 (resp. S_2) and $VIEW_1$ (resp. $VIEW_2$) are statistically indistinguishable. This concludes proof.

Protocol 5 Inverse of sum computation
Input: Alice and Bob have n and m, respectively
Output: Alice and Bob obtain r_1 and r_2, respectively

1: Alice chooses a random element $r \in F$ and defines the linear polynomial $Q_1(z) = rz + rn$.
2: Alice and Bob engage in a private evaluation of Q_1, in which Bob obtains $Q_1(m) = rm + rn$.
3: Bob chooses a random element $r_1 \in F$ and defines the linear polynomial $Q_2(z) = \frac{1}{rm+rn}z - r_1$
4: Alice and Bob engage in a private evaluation of Q_2, in which Alice obtains $r_2 = Q_2(r) = \frac{1}{n+m} - r_1$.
5: The respective outputs of Alice and Bob are defined as r_1 and r_2, giving us that $r_1 + r_2 = \frac{1}{n+m}$.

Proof of privacy: Similar to Protocol 4, Protocol 5 constitutes a private protocol for computing the inverse of sum as stated in Problem 2.

we give a sketch of proof for Protocol 5 as below:
The views of the two parties are

$$VIEW_1(n) = (n, r, \frac{1}{n+m} - r_1) = (n, \frac{1}{n+m} - r_1)$$

$$VIEW_2(m) = (m, rm + rn, r_1) = (m, rm + rn)$$

Let r_1' be an element uniformly chosen from F. An adversary cannot distinguish between $VIEW_1(n)$ and $(n,\ \frac{1}{n+m} - r_1')$ with more than negligible probability. Let r' and n' be elements uniformly chosen from F, it is straightforward that $VIEW_2(m)$ and $(m,\ r'm + r'n')$ are statistically indistinguishable. Therefore, privacy of party 1 with respect to party 2 follows.

IV. THE COMMUNICATION ANALYSIS

We give a analysis of the communication cost of the privacy preserving EM-base clustering algorithm (Protocol 1) at one iteration of the algorithm. 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, Protocol 4 and Protocol 5 are P_1, P_2, P_3, P_4 and P_5 respectively. We should note that the communication cost of the scalar product protocol is $O(n)$ (see in [9]), where n is size of input vectors and the communication cost of the oblivious polynomial evaluation protocol is $O(k)$ exponentiations [10] or $O(k|F|)$ (see in [11]), where k is the degree of the input polynomial. In one iteration of the algorithm, the communication occurs at steps 14, 18 and 21. Then, its the communication cost is $P_1 = m * P_3 + C * P_2$.

Protocol 3 is implemented by running one time the scalar product protocol that the size of input vectors is n, thus $P_3 = O(n)$. Protocol 2 implemented by running n times of Protocol 4 and one time of Protocol 5 (at step 1). At step 2, it runs n^2 times of the scalar product protocol with the input size be 6 and n^2 times with the input size be 4. Therefore, $P_2 = n * P_4 + P_5 + n^2 * (O(1) + O(1))$. Protocol 4 run three times of oblivious polynomial evaluation protocol, with the degree 1 polynomials, so $P_4 = 3 * O(|F|)$. Similarly, $P_5 = 2 * O(|F|)$. Finally, we have $P_1 = m * O(n) + 3 * C * n * O(|F|) + 2 * C * O(|F|) + 2n^2 * C * O(1))$. In fact, C and n are small, thus $P_1 \simeq O(n * m + C * n * |F|))$.

V. EXTEND TO THE K-MEANS ALGORITHM

In a iteration of the k-means algorithm, to determine the cluster of a object, the Euclidian distances between the object and the cluster centers need be computed. Thus, to preserve privacy for k-means clustering, the key step in this algorithm is privacy preserving of the cluster means. At each iteration of the algorithm, only means reveal to parties without other things, this method has presented in [4]. However, the problem is that revealing means might make privacy breaches, which was analyzed in section 4. In order to overcome this problem, we design a protocol, which allows each participating party to compute the distances between its objects and the cluster centers without revealing the cluster centers.

The distance (d) between the object x and the center of cluster k computed by the formula:

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

Assume that $\mu_k[i] = \mu_k^A[i] + \mu_k^B[i]$, we rewrite

$$d = (x[1] - \mu_k^A[1] - \mu_k^B[1])^2 + ... + (x[n] - \mu_k^A[n] - \mu_k^B[1])^2$$
$$= \sum_{k=1}^{n}(x[i] - \mu_k^A[1])^2 - 2\sum_{k=1}^{n}(x[i] - \mu_k^A[1])\mu_k^B[i] + \sum_{k=1}^{n}(\mu_k^B[i])^2$$

We note that

$$\sum_{k=1}^{n}(x[i] - \mu_k^A[1])\mu_k^B[i] =$$
$$(x[i] - \mu_k^A[1],\ ...,\ x[n] - \mu_k^A[n]) \times \begin{pmatrix} \mu_k^B[1] \\ \cdot \\ \cdot \\ \cdot \\ \mu_k^B[n] \end{pmatrix} = R^A \bullet R^B \quad (2)$$

Therefor, the privacy preserving the Euclidian distance protocol can be summarized as following

Two parties, Alice has C_k^A and the vector $x = (x[1],\ ...,\ x[n])$, Alice has C_k^B. Assume that $C_k = C_k^A \cup C_k^B$ is a cluster. Alice need to compute the distance between x and the center of cluster k. the algorithm is implemented based on four steps:

1: Alice and Bob jointly compute the means vector of C_k using Protocol 4.
2: Alice presents vector R^A, Bob present vector R^B, such as presented in (2)
3: Alice and Bob jointly compute the dot product of R^A and R^B using the scalar product. Alice obtains X^A, Bob obtains X^B.
4: Bob computes $Y^B = -2X^B + \sum_{k=1}^{n}(\mu_k^B)^2$ and send it to Alice
5: Alice computes $d = \sum_{k=1}^{n}(x[i] - \mu_k^A[1])^2 - 2X^A + Y^B$

VI. CONCLUSION

In this paper, we have presented algorithm to the problems of the privacy preserving EM-based clustering. Our algorithm allows two parties to cooperatively conduct clustering on their joint data sets without disclosing each party's private data to the other party. We have formulated two protocols for privacy preserving computation of means and inverse of sum, the proposed protocols are designed based on oblivious polynomial evaluation and proved secrecy based on the formal security model. The finally, we give the extension of the proposed method to address the problem of privacy preserving k-means clustering.

REFERENCES

[1] J. Han and M. Kamber. Data Mining Concepts and Techniques. Morgan Kaufmann Publishers, 2001.
[2] S. De Backer, P. Scheunders. A Competitive Elliptical Clustering Algorithm. in Pattern Recognition Letters, Vol. 20, Nr. 11-13, p. 1141-1147, (1999)

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

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

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

[6] International Conference on Data Mining, pages 222-233, 2004. W. Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In Proc. of the 17th Annual Computer Security Applications Confer- ence, pages 103110, 200

[7] 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.

[8] R. N. Wright and Z. Yang. Privacy-preserving Bayesian network structure computation on distributed heterogeneous data. In Proc. of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 713 718. ACM Press, 2004.

[9] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli 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.

[10] 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.

[11] 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.

[12] Jaideep Vaidya, Chris Clifton. Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data. Proceedings of the Fourth SIAM International Conference on Data Mining/ Michael W. Berry, Chandrika Kamath, Umeshwar Dayal, David Skillicorn, 2004.

[13] 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.

[14] 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.

[15] S. Jha, L. Kruger, P. McDaniel. Privacy Preserving Clustering. In ESORICS 2005.

[16] O. Goldreich. Foundations of Cryptography: Volume 1, Basic Tools. Cambridge University Press, May 2001.

[17] O. Goldreich. Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, 2004. A. Evfimevski, J.

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

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

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

[21] Andruszkiewicz, P. 2008. Probability Distribution Reconstruction for Nominal Attributes in Privacy Preserving Classification. In Proceedings of the 2008 international Conference on Convergence and Hybrid information Technology - Volume 00 (August 28 - 29, 2008)

[22] Agrawal, R., Srikant, R., and Thomas, D. 2005. 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.

[23] 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

[24] Kun Liu, Hillol Kargupta, and Jessica 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.

[25] Vassilios. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. ACM SIGMOD Record, 33(1), 2004

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

[27] Jaideep Vaidya, Chris Clifton. Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data. Proceedings of the Fourth SIAM International Conference on Data Mining/ Michael W. Berry, Chandrika Kamath, Umeshwar Dayal, David Skillicorn, 2004.

[28] *Zhu, Y. and Liu, L.* 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).

[29] 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. O.

[30] Goldreich, S. Micali, and A.Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. ACM Symp. on the Theory of Computing, 1987.

[31] Latanya 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.

[32] 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

[33] W. Du and M. J. Atallah, "Protocols for Secure Remote Database Access with Approximate Matching," 7th ACM Conference on Computer and Communications Security(ACMCCS 2000), 2000.

[34] Yan-Cheng Chang and Michael Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Cryptology ePrint Archive: Report 2004/051 J. Vaidya and C. Clifton.

[35] 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.

[36] Oliveira, S. R. M., Zaane, O. R. 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.

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

[38] Ali ..nan, Ycel Saygn, Erkay Sava.., Aya Azgn Hinto..lu, Albert Levi. Privacy Preserving Clustering on Horizontally Partitioned Data. Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)[7]