CHAPTER 3

OVERVIEW OF SECURED MULTIPARTY COMputation algorithms

The major focus in several distributed methods for privacy-preserving data mining is to permit computation of needful aggregate statistics over the entire data set without losing the privacy of the individual data sets within the different participants. Hence, the participants like to collaborate for getting aggregate results. They may not fully trust fellow participants in terms of distribution of their own data sets. For this reason, the data sets may either be horizontally partitioned or be vertically partitioned. Both kinds of partitioning offer different challenges to the problems encountered in distributed privacy preserving data mining. The problem of distributed privacy-preserving data mining overlaps closely with a field in cryptography so as to find out a secure multi-party computation. A wider outline of the intersection between the cryptography and privacy-preserving data mining fields may be observed in Pinkas (2002).

A major approach in cryptographic methods tends to compute functions over inputs offered by multiple recipients without actually sharing the inputs with each another. A 2-party setting is taken for example. Alice and Bob may have x and y inputs respectively and may wish to compute both the function \( f(x, y) \) without disclosing x or y to each of them. This difficulty encountered is generalized across k parties by designing the k argument function \( h(x_1 \ldots x_k) \). Several data mining algorithms can be seen in the context of repetitive computations of many such primitive functions like scalar dot product, secure sum etc. For to compute the function \( f(x, y) \) or
h(x_1, ..., x_k), a protocol has to be designed to exchange information in such a manner that the function is computed without anything being compromised in privacy. It is observed that the robustness of the protocol is based on level of faith one is willing to keep on the two participants Alice and Bob. This is due to the fact that the protocol may be subjected to several types of adversarial behavior like semi-honest and malicious adversaries.

Several techniques are employed in data mining primitives to secure multi-party computation over different horizontally and vertically partitioned data sets. Some of the effective algorithms meant for secure multi-party computation over horizontally and vertically partitioned data sets are discussed next. The remaining chapter is arranged as follows; Section 3.1 talks about fundamental cryptographic techniques used for PPDDM. It is followed by explanation of some popular methods used in literature for horizontally and vertically partitioned data in Section 3.2 and 3.3 respectively. The chapter ends with the shortcomings and problems found in the existing algorithms in Section 3.4.

3.1 BASIC CRYPTOGRAPHIC TECHNIQUES FOR PPDDM

Privacy-preserving distributed data mining algorithms are needed for collaborating between parties to compute the results, when possibly protecting disclosure of any information other than the data mining results. To accomplish this task, tools from Secure Multiparty Computation (SMC) are employed. Privacy as a concept in this approach, depends on a solid body of theoretical work. Here, initially some basic ideas from SMC domain are discussed. They are followed by description of a useful variant of public-key cryptography system called homomorphic encryption. The SMC literature gives definitions for two basic adversarial models:
• Semi-Honest: Semi-honest (or Honest but Curious) adversaries faithfully follow the protocol. At the same time, they take efforts to infer the secret information of other parties from the data found during execution of the protocol.

• Malicious: To infer secret information, malicious adversaries will do anything. This includes aborting the protocol at any time, send spurious messages, spoof messages, collude with other (malicious) parties, etc.

3.2 PPDDM ON HORIZONTALLY PARTITIONED DATA

Here, this section presents an outline of some of the popular PPDDM algorithms on horizontally partitioned data. In the concluding part of the chapter, there is a brief discussion on the challenges in privacy-preserving algorithms developed for horizontally partitioned data.

ID3 Decision Tree Mining

It is the starting work on privacy-preserving distributed data mining on horizontally partitioned data (Lindell & Pinkas 2000). The target is to build a secure ID3 decision tree. Here, the training set will be horizontally distributed between two parties. The basic concept is in identifying the attribute that enhances information gain which is equivalent in identifying the attribute that enhances the conditional entropy. The conditional entropy for an attribute between two parties is written as a sum of the expression of the form \((v_1 + v_2) \times \log(v_1 + v_2)\). Secure log algorithm are used by the authors to secure polynomial evaluation and comparison sub-protocols for calculating the above mentioned expression and show the way to apply their function in building ID3 in a secure manner.
**Association Rule Mining**

Here, the aim of privacy-preserving association rule mining is in framing compute rules in the form $X \Rightarrow Y$ (e.g. Bread implies Milk) which enjoys a global support and confidence over some certain threshold. It is well proved in Kantarcioglu & Clifton (2004). It can be attained by employing a secure set union, secure summation and secure comparison sub-protocols. The algorithm explained in Kantarcioglu & Clifton (2004) contains two phases. The first phase employs secure set union to obtain a union of candidate association rules. In the next phase, secure summation and secure comparison are employed to filter the candidate items which are not globally supported.

**Naive Bayes Classification**

The Naive Bayes classifier is a highly practical Bayesian learning method, which is used in learning tasks where each instance $x$ is explained by a conjunction of attribute values and the target function $f(x)$ which can take on any value from some finite set $C$. In Naive Bayes classification, for classifying an instance represented as a tuple of attribute values $<a_1, a_2, ..., a_n>$, it is imperative to measure the conditional probabilities $P(a_i|c_j)$ for all $c_j \in C$ by employing the training set. The prior probabilities $P(c_j)$ for all $c_j \in C$ is necessary to be fixed in some fashion (by simply counting the frequencies from the training set). The probabilities for differing hypotheses (classes) may computed by normalizing the values required from each hypothesis (class). Kantarcioglu et al (2003) exhibit that by computing $P(a_i|c_j)$, it is possible to securely reduce computing a function of the form

$$\frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} y_i}$$
Here $x_i$, $y_i$ values are known by site $i$. At the same time, it is possible to securely calculate by using secure summation and secure $\ln(x)$ protocol.

**k-NN Classification**

It predicts the class value of an instance by employing the k nearest examples which depend in the training data. Several distance metrics are employed to determine the k nearest examples. Kantarcioglu & Clifton (2004) suggest about a privacy preserving k-NN algorithm based on the assumption that the instance necessitates to be classified as public. The approach employs an untrusted, non-colluding party that is not permitted to learn anything about any of the data, but it is believed in not colluding with other parties to reveal private information. The simple idea is that each site locates its own k-nearest neighbors, (this is possible since the instance that needs to be classified is public and the data is horizontally partitioned). This encrypts the class by using the public key of the site which sends the instance for classification (querying site).

The parties compare securely their k-nearest neighbors with all other sites. This is possible except that the comparison gives each site a random share of the result and hence no party learns the result of the comparison. The results obtained from all sites are combined, scrambled, and given to the untrusted, non-colluding site. This site combines the random shares to obtain the comparison result for each pair. Thus, it enables it to sort and select the global k-nearest neighbors (but without learning the source or values of the items). The querying site and the untrusted, non-colluding site make use of a protocol to identify the class value.
Support Vector Machine Classification

Support Vector Machine (SVM) classification is another vital classification technique. Yu et al (2006) offers a privacy-preserving solution for horizontally partitioned case by employing a secure dot product sub-protocol. The solution offered in Yu et al (2006) employs the observation to construct the SVM. For this, only the kernel matrix $K$ is needed. To calculate the Kernel matrix $K$, the gram matrix $G$ where $G_{ij} = x_i \cdot x_j$ is necessary to be computed securely for all training instance pairs $x_i, x_j$. Clearly $G_{ij}$ can be calculated by employing secure dot-product protocol.

k-means and EM Clustering

Clustering is a well-studied data mining technique that attempts to group similar instances in a given data set into clusters for minimizing some objective function. In k-means clustering, the aim is to partition data into $k$ clusters. Normally, $k$ initial cluster centers are selected, and then the condition of the cluster centroids are updated by applying an iterative method. Jagannathan & Wright (2005) exhibit that it is possible to achieve k-means clustering by employing secure dot product, secure summation and secure comparison. Similarly, Lin et al (2005) give definition for a secure clustering by applying the expectation maximization method for horizontally partitioned data by applying a secure summation protocol.

3.3 PPDDM ON VERTICALLY PARTITIONED DATA

It is commonly observed among vertically partitioned data that the presence of more data considerably enhances the quality of the models built from the dataset. On the whole, the data analysis results are found to be significantly real and useful. While this is the case with horizontally partitioned data (more data is always good), yet it gives a more critical impact
with vertically partitioned data. This is due to the fact that the data obtained from different parties produce significantly different extra information about the entities. The complex nature of privacy-preserving data mining is very much improved because of vertical partitioning data. Quite contrary to horizontal partitioning of data, vertical partitioning of data gives raise to several unique questions in connection to the manner in which data is processed. The results are obtained and shared. This section makes a survey of different types of privacy-preserving data mining algorithms following the main data mining tasks of association rule mining, classification, clustering and outlier detection. The tasks are quickly studied and completed with the complication that is specific to vertical partitioning of data and some observed inherent challenges.

Classification

It refers to the problem in categorizing observations into classes. Predictive modeling employs samples of data for which the class is known to produce a model in classifying new observations. One issue with classification for vertically partitioned data is whether the class attribute is shared among all parties or is local to only one of them. By having the class attributes which are known to all parties makes the problem simple. However, it may not always be the case. When the class attribute is known to only one party, any process that necessitates in counting the number of entities having a particular value for an attribute and a particular class will have to be secure. It means that computing information gain, etc. should be completely secure. Another issue with classification is classification model shared among parties? One possibile thing is to allow all the parties know the developed model, but it may reveal too much information. The totally secure alternative is to retain the created model completely split between the parties. However, it can have significant impact on the classification time. Other alternatives are
also possible with differing tradeoffs between security and cost. The rest of
the sub-section discusses some of the popular classifiers.

**Naive Bayes Classification**

It is simple but very much effective classifier. Its simplicity and
effectiveness has paved way for being used as a baseline standard by which
other classifiers are calculated. Vaidya & Clifton (2004) offer a privacy
preserving solution for vertically partitioned data. The Naive Bayes classifier
is employed to learning tasks where each instance $x$ is explained by a
conjunction of attribute values and the target function $f(x)$ is able to take any
value from some finite set $C$.

**Bayesian Network Structure Learning**

These Networks make the attribute independence assumption of the
Naive Bayes classifier, very much flexible by capturing situations. Here, the
dependencies between attributes affect the class. A Bayesian Network is a
graphical model. In this, the vertices correspond to attributes, and the edges
maintain probabilistic relationships between the attributes (Naive Bayes is
thus a Bayesian Network with no edges.) Wright & Yang (2004) suggest a
privacy-preserving protocol to learn about the Bayesian network structure for
vertically partitioned data. This protocol is narrowed between two parties. The
fundamental approach is to emulate the K2 algorithm (Cooper & Herskovits
1992) starts with a graph containing no edges. It later chooses a node and
reluctantly adds a “parent” edge to that node that mostly improves a score for
the network. It there by stops when a threshold for number of parents is
reached.
Decision Tree Classification

It offers a solution to construct ID3 on vertically partitioned data. It was proposed by Du & Zhan (2002). The work here is carried out by assuming that the data is vertically partitioned between two parties. The class of the training data is believed to be shared, but some of their attributes are private. Hence many steps of the ID3 algorithm may be evaluated locally. The major problem is in computing the site that has the best attribute to split on – each can compute the gain of their own attributes without reference to the other site. Vaidya et al (2008) offers a solution which solves a more general problem. It is done by creating an ID3 decision tree when the training data is vertically partitioned between many parties (≥ 2) and only a single party knows about the class attribute. As each party has knowledge about only some attributes, knowing the structure of the tree (especially, knowledge of an unknown attribute and its breakpoints for testing). It enables in violation of privacy among individual parties so as to ensure zero leakage of extra information. Here there is a necessity to the structure of the tree by using an oblivious protocol to classify a new instance. However, its cost is not acceptable and hence a compromise is arrived by hiding the attribute tests used in the tree while revealing the basic structure of the tree.

Clustering

Sharing the clusters in PPDDM is a challenging task. Specifically, it is only cluster membership shared or is more information about the clusters shared. In the later case, other parties can easily learn a lot of information about the other attributes. Vaidya et al (2008) offered a method for clustering over vertically partitioned data – a privacy-preserving protocol performs do k-means clustering. Although all parties are aware of the final assignment of data points to clusters, they withhold only portion of information for each cluster. The cluster centers $\mu_i$ are believed to be semiprivate information, i.e.,
each site can get only a few components of $\mu$ that match with attributes it holds. Hence, all information about a site’s attributes (not just individual values) is maintained confidentially. If there is a willingness to share $\mu$, then there is a possibility to evaluate of privacy/secrecy after the values are known.

The basic protocols suggested adopt the original K-means protocol. There are two big problems encountered – figuring out how to assign points to clusters in each iteration, and to know when to stop. As the means at each iteration are not taken into consideration for private information, figuring out when to stop is simple. Each party can locally compute the difference between their shares of the mean, and finally verify whether the total difference is fewer than the threshold. As all arithmetic is carried out, the threshold evaluation carried out at the end is not transparent. Intervals are compared rather than the actual numbers. More details can be seen in Vaidya et al (2008). Assigning points to clusters in each iteration is done through a secure protocol by employing three key ideas:

i. Disguise the site components of the distance by using random values that get cancelled when combined.

ii. Compare distances so that its result is observed. No one is aware of the distances being compared.

iii. Permute the order of clusters in such a way that the real meaning of the comparison results is not known.

One disadvantage of the Vaidya and Clifton protocol is that it is not totally secure as intermediate results are revealed. Necessarily, the intermediate cluster assignment of data points is made known to every party for each iteration, although the end result only specifies the final clusters. But, this compromise is necessary for efficiency. Jagannathan & Wright (2005) offer a totally secure protocol for arbitrarily partitioned data. The protocol
used is quite identical to the Vaidya and Clifton protocol by having the added complexity of splitting the intermediate cluster centers. Hence, no information is leaked whatsoever.

**Association Rule Mining**

Vaidya & Clifton (2002) first exhibited how it is possible to apply secure association rule mining that can be done for vertically partitioned data by broadening the apriori algorithm. Vertical partitioning means that an itemset can be split among multiple sites. Several steps of the apriori algorithm are carried out locally at each of the sites. The important step involved is in identifying the support count of an itemset. When the support count of an itemset is securely computed, it is possible to check whether the support is greater than threshold, and decide about the frequent occurrence of itemset. By employing this, the association rules may easily become secure. The major concept of Vaidya & Clifton (2002) is in computing the support of an itemset exactly using the scalar product of the vectors representing the sub-itemsets with different parties. Hence, the complete secure association rule mining problem may be minimized in computing the scalar product of two vectors in a privacy-preserving manner. Vaidya & Clifton (2002) suggested an algebraic method to compute the scalar product. Though this method is not totally secured, yet it is very much efficient. The strength of secure association rule mining protocol is in being not tied to any specific scalar product protocol. There are several secure scalar product protocols proposed. Here, at least two are completely secure. All have differing tradeoffs of security, efficiency, and utility (some are limited to scalar products over Boolean data). Kantarcioglu (2008) introduces one possible secure protocol for computing the scalar product by employing homomorphic encryption. Although several solutions are obtained by employing scalar product computation, one alternative solution is needed to be mentioned.
Vaidya & Clifton (2005) offer an innovative alternative solution for association rule mining problem. Two key insights are offered in this solution. First is when the vectors as sets (with position numbers as elements) are encoded. Here, the scalar product is the same as the size of the intersection set. The fundamental idea is to employ commutative encryption so as to encrypt all the items in each party’s set. Commutative encryption is a vital tool in many cryptographic protocols. An encryption algorithm is commutative when the order of encryption does not matter. Hence, for any two encryption keys $E_1$ and $E_2$, and any message $m$, the $E_1(E_2(m)) = E_2(E_1(m))$. The same property is applied to decryption also. For decrypting a message encrypted by two keys, it is enough to decrypt one key at a time. The concept is for each source to encrypt its data set with its keys and to transmit the encrypted data set to the next source. This source again encrypts the received data by employing its encryption keys and transmits the encrypted data to the next source until all sources have encrypted the data. As these methods are employing commutative encryption, the encrypted values of the set items across various data sets should be equal when the original values are equal. Hence, all intersection of the encrypted values offer the logical AND of the vectors and measuring the size of the intersection set provides the total number of 1’s (i.e., the scalar product). The encryption protects any party from becoming aware of the actual value of any local item. This scalar product method works only with Boolean vectors, yet it still works with association rule mining problem. Their concept is employed by Agrawal et al (2003) to compute Set Union, Set Intersection, Size of Set Union, and Size of Set Intersection. However their work is confined to two parties. Freedman et al (2004) suggest techniques by employing homomorphic encryption to do private matching and set intersection for two parties which is capable of guarding against malicious adversaries in random oracle model also. Though it is a good alternative, its real innovativeness is found in realizing that all the items are encrypted by the keys of all the parties. They are able to locally
compute all the frequent itemsets. It means that the overall cost of secure association rule mining is nothing but the cost of completely encrypting all the items. When there are k parties, n items and m transactions, the entire cost of association rule mining is \( O(nmk) \). As there will be total number of encryptions required (the encryption time dominates all other costs), it is to be noted that it is independent of the number of frequent itemsets that may easily be in tens of thousands. Therefore, the protocol in Vaidya & Clifton (2005) is very much efficient in global sense and it creates privacy-preserving association rule mining in a most feasible manner.

Several protocols created assume a semi-honest model. Here, the parties involved will follow the protocol honestly, yet they may later try to infer additional information from data they receive through the protocol. One result obtained from it is that parties are not allowed to offer spurious input to the protocol. When a party is permitted to offer spurious input, they may probe to find out the value of a specific item from other parties. Say for example, when a party gives the input \((0, ..., 0, 1, 0, ..., 0)\), the result of the scalar product \((1 \text{ or } 0)\) talks to the malicious party regarding the other party whether the transaction corresponds to 1. Attacks of this kind may be called probing attacks and they should be protected from such attacks. The protocol in Vaidya & Clifton (2005) may protect from such attacks partially.

**Outlier detection**

Outlier/anomaly detection is a common data mining task which is there in practice. Hawkins (1980) gives definition about it as an observation which deviates from other observations to arouse suspicion which is created by a different mechanism. Outlier detection is employed to identify uncommon sequences in general data, for finding fraudulent transactions in credit card records, fraud discovery in mobile phones, as well as to find intrusions from network traffic data (Lazarevic et al 2003), etc. It is known
for detecting previously unknown suspicious behavior as it is a clear outlier detection problem. Several applications have privacy concerns, and organizations should be careful in avoiding the overstepping bounds of privacy legislation.

3.4 CHALLENGES OF THE CRYPTOGRAPHIC TECHNIQUES

The study says that there are quite a few constraints in cryptographic approaches employed in PPDDM as shown below:

- The cryptographic operations need higher overhead.

- Parameters like support threshold, secure sum should be carefully selected.

- By employing attributes with large number of discrete values, there is a necessity for higher computation times.

- The privacy needs to be examined carefully before applying privacy preserving distributed data mining protocols.

- In comparison with noise addition methods, the cryptographic techniques will not allow easy trade-off between privacy and accuracy. Feigenbau et al (2006) offer a good starting point in that direction.

- Not suitable for “rational” parties participating in the protocols. Jiang et al (2008) state that when the participating parties are rational, cost reductions in the malicious model can be achieved significantly.

- For vertically partitioned data, it is very challenging to carry out many local aggregations beforehand.
In this chapter, a survey of efficient solutions for several privacy preserving data mining tasks on horizontally and vertically partitioned data are presented. Quite similar to horizontally partitioned data, it is observed that even for vertically partitioned data, many privacy-preserving algorithms may be efficiently implemented by bringing together specific basic secure building blocks. As data mining is typically carried out for millions of transactions, its cost goes up significantly. Hence, there is a necessity for efficient protocols to be implemented. This necessitates the change in ensuring deployment of these algorithms into real life. On the whole, there is a positive trend towards privacy-preserving algorithms. Because of increase in privacy and security concerns along with the need for leverage commercial assets, there is a felt necessity for flexible and efficient privacy-preserving solutions that can be used for individual privacy needs. Growth of such flexible and efficient solutions may be instrumental in adopting this technology in a large scale.