International Journal of Research
Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 04
February 2016

Privacy in Distributed Database

Priyanka Pandey[1], Javed Aktar Khan[2]
[1,2]Department of Computer Science and Engineering, Takshshila Institute of Engineering and Technology, Jabalpur, M.P., India
pandeypriyanka906@gmail.com; javedaktarkhan@takshshila.org

Abstract—

Privacy concern often constraint data mining. This paper addresses the problem of Association rule mining where operation is distributed across multiple sites. Each site holds the some transaction, and the sites wish to collaborate to identify valid Association rule globally. However the sites must not acknowledge individual data. We presents the multi party transaction data who discovering frequent item sets with minimum support, without either sites knowing the individual data.

Keywords- Horizontal Partitioning; Association Rule Mining; Secure multi party computation; Ck secure sum; modified Ck secure sum; Rk Secure Sum; Modified Rk Secure Sum.

1. Introduction

Association rule mining (ARM) has become one of the core data mining tasks and has attracted tremendous interest among data mining researchers. ARM is an undirected or unsupervised data mining technique which works on variable length data, and produces clear and understandable results. There are two dominant approaches for utilizing multiple processors that have emerged distributed memory in which each processor has a private memory; and shared memory in which all processors access common memory. Shared memory architecture has many desirable properties. Each processor has direct and equal access to all memory in the system. Parallel programs are easy to implement on such a system. In distributed memory architecture each processor has its own local memory that can only be accessed directly by that processor. For a processor to have access to data in the local memory of another processor a copy of the desired data element must be sent from one processor to the other through message passing. XML data are used with the Optimized Distributed Association Rule Mining Algorithm. A parallel application could be divided into number of tasks and executed concurrently on different processors in the system. However the performance of a parallel application on a distributed system is mainly dependent on the allocation of the tasks comprising the application onto the available processors in the system.

Modern organizations are geographically distributed. Typically, each site locally stores its ever increasing amount of day-to-day data. Using centralized data mining to discover useful patterns in such organizations' data isn't always feasible because merging data sets from different sites into a centralized site incurs huge network communication costs. Data from these organizations are not only distributed over various locations but also vertically fragmented, making it difficult if not impossible to combine them in a central location. Distributed data mining has thus emerged as an active subarea of data mining research. In this paper an Optimized Association Rule Mining Algorithm is used for performing the mining process.

2. Association Rule Mining Problem

With the general example and introduction in last section, the formal statement of association rule mining problem was firstly stated in [1] by Agrawal. Let I=I_1, I_2, ……., I_m be a set of m distinct attributes, T be transaction that contains a set of items such that T sub set of I, D be a database with deferent transaction records Ts. An association rule is an implication in the form of X=>Y, where X, Y sub set of I are sets of items called item sets, and $X \cap Y = ¢$. X is called antecedent while Y is called consequent, the rule means X implies Y.

There are two important basic measures for association rules, support(s) and Confidence(c). Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful. The Two thresholds are called minimal support and minimal confidence respectively, Additional constraints of interesting rules also can be specified by the users. The two basic parameters of Association Rule Mining (ARM) are: support and confidence.

Support(s) of an association rule is defined as the percentage/fraction of records that contains $X \cup Y$ to the total number of records in the database. The count for each item is increased by one every time the item is encountered in different transaction T in database D during the scanning process. It means the support count does not take the quantity of the item into account. For example in a transaction a customer buys three bottles of beers but we only increase the support count number of {beer} by one, in another word if a transaction contains a item then the support count of this item is increased by one. Support(s) is calculated by the following
Formula:

Support (X∪Y) = Support count of X∪Y / Total number of transaction in D

From the definition we can see, support of an item is a statistical significance of an association rule. Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contains purchasing of this item. The retailer will not pay much attention to such kind of items that are not bought so frequently, obviously a high support is desired for more interesting association rules. Before the mining process, users can specify the minimum support as a threshold, which means they are only interested in certain association rules that are generated from those item sets whose supports exceed that threshold. However, sometimes even the item sets are not so frequent as defined by the threshold, the association rules generated from them are still important. For example in the supermarket some items are very expensive, consequently they are not purchased so often as the threshold required, but association rules between those expensive items are as important as other frequently bought items to the retailer.

Confidence(c) of an association rule is defined as the percentage/fraction of the number of transactions that contain XUY to the total number of records that contain X, where if the percentage exceeds the threshold of confidence an interesting association rule X=>Y can be generated. Formula:

Confidence(X∪Y) = Support(X∪Y) / Support(X)

Confidence is a measure of strength of the association rules, suppose the confidence of the association rule X=>Y is 80%, it means that 80% of the transactions that contain X also contain Y together, similarly to ensure the interestingness of the rules specified minimum confidence is also pre-defined by users.

2.1 Apriori Algorithm

An association rule mining algorithm, Apriori has been developed for rule mining in large transaction databases by IBM's Quest project team. An item set is a non-empty set of items.

They have decomposed the problem of mining association rules into two parts

1. Find all combinations of items that have transaction support above minimum support. Call those combinations frequent item sets.
2. Use the frequent item sets to generate the desired rules. The general idea is that if, say, ABCD and AB are frequent item sets, then we can determine if the rule AB CD holds by computing the ratio r=support(ABCD)/support(AB). The rule holds only if r >= minimum confidence. Note that the rule will have minimum support because ABCD is frequent. The algorithm is highly scalable .The Apriori algorithm used in Quest for finding all frequent item sets is given below

```
Procedure Apriori Algo( )
Begin
    L1:= {frequent 1-itemsets};
    for ( k := 2; $L_{k-1}$; k++ ) do {
        Ck= apriori-gen($L_{k-1}$) ; // new candidates for all transactions t in the dataset
        Do
        {
            for all candidates c Ck contained in t do
                c: count++
        }
        $L_k$ = {c $C_k$ | c:count >= min-support}
    }
    Answer := k $L_k$
End
```

It makes multiple passes over the database. In the first pass, the algorithm simply counts item occurrences to determine the frequent 1-itemsets (item sets with 1 item). A subsequent pass, say pass k, consists of two phases. First, the frequent item sets Lk-1 (the set of all frequent (k-1)-item sets) found in the (k-1) th pass are used to generate the candidate item sets Ck, using the apriori-gen() function. This function first joins Lk-1 with Lk-1, the joining condition being that the lexicographically ordered first k-2 items are the same. Next, it deletes all those item sets from the join result that have some (k-1)-subset that is not in Lk-1 yielding Ck. The algorithm now scans the database. For each transaction, it determines which of the candidates in Ck are contained in the transaction using a hash-tree data structure and increments the count of those candidates. At the end of the pass, Ck is examined to determine which of the candidates frequent, yielding Lk is. The algorithm terminates when L_k becomes empty.

2.2 Distributed Association Ruling

DARM discovers rules from various geographically distributed data sets. However, the network connection between those data sets isn't as fast as in a parallel environment, so distributed mining usually aims to minimize communication costs. Researchers proposed the Fast Distributed Mining algorithm to mine rules from distributed data sets partitioned among different sites.3, in each site, FDM finds the local support counts and prunes all infrequent local support counts. After completing local pruning, each site broadcasts messages containing all the remaining candidate sets to all other sites to request their support counts. It then decides whether large item sets are globally frequent and generates the candidate item sets from those globally frequent item sets.

FDM's main advantage over CD is that it reduces the communication overhead to O (|Cp|*n), where |Cp| and n are potentially large candidate item sets and the number of sites, respectively. FDM generates fewer candidate item sets compared to CD, when the number of disjoint candidate item sets among various sites is large. However, we can only achieve this when different sites have non homogeneous data sets. FDM's message optimization techniques require some functions to determine the polling site, which could cause extra computational cost when each site has numerous local frequent item sets. Furthermore, each polling site must send a request to remote sites other than the originator site to find an item set's global support counts, increasing message size when numerous remote sites exist.

Recently, Assaf Schuster and his colleagues proposed the Distributed Decision Miner,14 which reduces communication overhead to O (Pr_{above} * |C| * n), where Pr_{above} is the probability of a candidate item set that has support greater than the support threshold. It generates only those rules that have confidence above the threshold level without generating a rule's exact confidence, therefore considering all rules above the confidence threshold as being the same. However, ARM is an iterative process, and it's hard for an algorithm to guess a priori how many rules might satisfy a given level of support or confidence. Furthermore, the final rule model this approach generates won't be identical at different sites because it generates rules using an item set's partial support count.

2.3 Privacy in Data Mining

The problem of privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms to leverage this information. A number of techniques such as randomization and k-anonymity [1] [4] [16] have been suggested in recent years in order to perform privacy-preserving data mining. Furthermore, the problem has been discussed in multiple communities such as the database community, the statistical disclosure control community and the cryptography community. In some cases, the different communities have explored parallel lines of work which are quite similar. This book will try to explore different topics from the perspective of different communities, and will try to give a fused idea of the work in different communities. The key directions in the field of privacy-preserving data mining are as follows:

1. **Privacy-Preserving Data Publishing:** These techniques tend to study different transformation methods associated with privacy. These techniques include methods such as randomization [1], k-anonymity [7] [16], and l-diversity [11]. Another related issue is how the perturbed data can be used in conjunction with classical data

mining methods such as association rule mining [15]. Other related problems include that of determining privacy-preserving methods to keep the underlying data useful (utility-based methods), or the problem of studying the different definitions of privacy, and how they compare in terms of effectiveness in different scenarios.

2. **Changing the Results of Data Mining Applications to Preserve Privacy:** In many cases, the results of data mining applications such as association rule or classification rule mining can compromise the privacy of the data. This has spawned a field of privacy in which the results of data mining algorithms such as association rule mining are modified in order to preserve the privacy of the data. A classic example of such techniques are association rule hiding methods, in which some of the association rules are suppressed in order to preserve privacy.

3. **Query Auditing:** Such methods are akin to the previous case of modifying the results of data mining algorithms. Here, we are either modifying or restricting the results of queries. Methods for perturbing the output of queries are discussed in [8], whereas techniques for restricting queries are discussed in [9] [13].

4. **Cryptographic Methods for Distributed Privacy:** In many cases, the data may be distributed across multiple sites, and the owners of the data across these different sites may wish to compute a common function. In such cases, a variety of cryptographic protocols may be used in order to communicate among the different sites, so that secure function computation is possible without revealing sensitive information. A survey of such methods may be found in [14].

5. **Theoretical Challenges in High Dimensionality:** Real data sets are usually extremely high dimensional and this makes the process of privacy preservation extremely difficult both from a computational and effectiveness point of view. In [12], it has been shown that optimal kanonymization is NP-hard. Furthermore, the technique is not even effective with increasing dimensionality, since the data can typically be combined with either public or background information to reveal the identity of the underlying record owners. A variety of methods for adversarial attacks in the high dimensional case are discussed in [5] [6]. This book will attempt to cover the different topics from the point of view of different communities in the field. This chapter will provide an overview of the different privacy-preserving algorithms covered in this book. We will discuss the challenges associated with each kind of problem, and discuss an overview of the material in the corresponding chapter.

2.4 Privacy Preservation Techniques and Protocols

The problem of privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms to leverage this information. A number of techniques such as randomization and k-anonymity have been suggested in recent years in order to perform privacy-preserving data mining. Furthermore, the problem has been discussed in multiple communities such as the database community, the statistical disclosure the different communities have explored parallel lines of work which are quite similar. This book will try to explore different topics from the perspective of different communities, and will try to give a fused idea of the work in different communities. Control community and the cryptography community. In some cases, the key directions in the field of privacy-preserving data mining are as follows:

1. **Privacy-Preserving Data Publishing:** These techniques tend to study different transformation methods associated with privacy. These techniques include methods such as randomization, k-anonymity, and l-diversity. Another related issue is how the perturbed data can be used in conjunction with classical data mining methods such as association rule mining. Other related problems include that of determining privacy-preserving methods to keep the underlying data useful (utility-based methods), or the problem of studying the different definitions of privacy, and how they compare in terms of effectiveness in different scenarios.
2. **Changing the results of Data Mining Applications to preserve privacy:** In many cases, the results of data mining applications such as association rule or classification rule mining can compromise the privacy of the data. This has spawned a field of privacy in which the results of data mining algorithms such as association rule mining are modified in order to preserve the privacy of the data. A classic example of such techniques are association rule hiding methods, in which some of the association rules are suppressed in order to preserve privacy.
3. **Query Auditing:** Such methods are akin to the previous case of modifying the results of data mining algorithms. Here, we are either modifying or restricting the results of queries. Methods for perturbing the output of queries are discussed in, whereas techniques for restricting queries are discussed in.
4. **Cryptographic Methods for Distributed Privacy:** In many cases, the data may be distributed across multiple sites, and the owners of the data across these different sites may wish to compute a common function. In such cases, a variety of cryptographic protocols may be used in order to communicate among the different sites, so that secure function computation is possible without revealing sensitive information. A survey of such methods may be found in.
5. **Theoretical Challenges in High Dimensionality:** Real data sets are usually extremely high dimensional and this makes the process of privacy preservation extremely difficult both from a computational and effectiveness point of view. In, it has been shown that optimal k-anonymization is NP-hard. Furthermore, the technique is not even effective with increasing dimensionality, since the data can typically be combined with either public or background information to reveal the identity of the underlying

record owners. A variety of methods for adversarial attacks in the high dimensional case are discussed in.

2.4.1 Ck-Secure Sum Protocol

In k-secure sum protocol [4] a middle party can be hacked by two neighbor parties with some probability. The technique for ck-Secure Sum Protocol [3] is that we change the neighbors in each round of segment computation. Thus it is guaranteed that no two semi honest parties can know all the data segments of a victim party. In this protocol each of the parties breaks the data block into k = n-1 segments where n is the number of parties involved in secure sum computation. We select P1 as the protocol initiator. The position of the protocol initiator is kept fixed in all the rounds of computation. For the first round of the computation parties are arranged serially as P1, P2,…., Pn.

The protocol initiator starts computation to get the sum of first segments of each party. For this computation our k-Secure Sum protocol [4] is used. Now, P2 exchanges its position with P3 and second round of computation is performed. Now, P2 exchanges its position with P4 and so on. Formally, in ith round of the computation P2 exchanges its position with Pi+1 until Pn is reached. In each round of computation, segments are added and the partial sum is passed to the next party until all the segments are added. Finally, the sum is announced by the protocol initiator party. The Ck-Secure Sum Protocol provides privacy against two colluding neighbors. Its analysis shows that when more than two parties collude, they can know the data of some party. The protocol initiator can be attacked by more than two parties that maliciously cooperate to know secret data of the protocol initiator. But for that also a specific combination of the parties must join against the protocol initiator. Any party who moves its position cannot be attacked by any group of the parties.

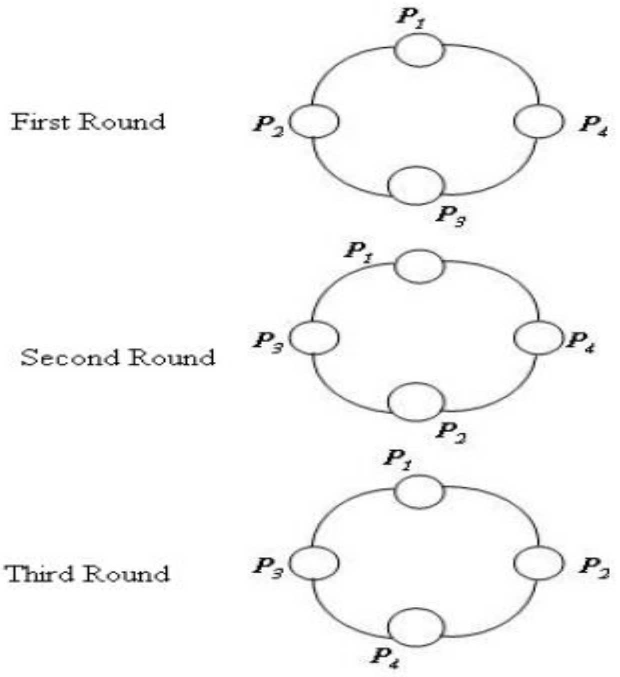

Figure1 Ck secure sum process

2.4.2 Modified Ck-Secure Sum Protocol

In this protocol mainly two modifications to the Ck-Secure Sum Protocol are done:

1. The number of segments k is kept equal to the number of the parties' n.
2. The protocol initiator party moves through the ring.

All the parties are arranged in a unidirectional ring. Each party divides its data block into k segments which is equal to the number of the parties. The party P**1** is selected as the protocol initiator party for all the rounds of the computation. This party starts computation by sending first data segment to the next party in the ring. The next party adds its segment to the received segment and the sum is passed to the next party. This process continues until all the segments are added. After receiving the sum of the first segments of all the parties, the protocol initiator P**1** exchanges its position with P**2** and then it sends the sum of its segment and the previous received sum to the next party in the newly arranged ring. At the end of this round, the protocol initiator receives the sum of two segments of all the parties. Now, P**1** exchanges its position with P**3** and so on until P**n** is reached.

2.4.3 Dk-Secure Sum Protocol

Assume, P1, P2 , …, Pk are k parties involved in cooperative secure sum computation where each party is capable of breaking its data block into a fixed number of segments such that the sum of all the segments is equal to the value of the data block of that party. In proposed protocol number of segments in a data block is kept equal to the number of parties. The values of the segments are randomly selected by the party and it a secret of the party. If k be the number of segments (which is same as the number of parties) then in this scheme each party holds any one segment with it and k-1 segments are sent to k-1 parties, one to each of the parties. Thus at the end of this redistribution each of the parties holds k segments in which only one segment belongs to the party and other segments belong to remaining parties, one from each. In this protocol, one of the parties is unanimously selected as the protocol initiator party which starts the computation by sending the data segment to the next party in the ring. The receiving party adds its data

Segment to the received partial sum and transmits its result to the next party in the ring. This process is repeated until all the segments of all the parties are added and the sum is announced by the protocol initiator party.

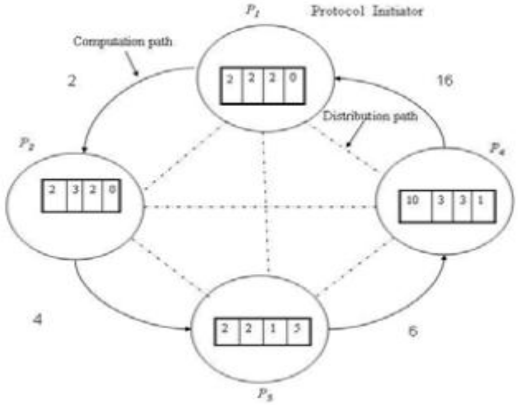

Figure 2 Dk secure sum process

Now even if two adjacent parties maliciously cooperate to know the data of a middle party they will be able to know only those k segments of a party which belong to every party.

The sum of these segments is a garbage value and thus worthless for the hacker party.

2.4.4 Rk Secure Sum Protocol

Let P_1, $P_{2,}$…..,P_k are k parties concerned in mutual secure sum computation where each party is accomplished of breaking its data block into a fixed number of data segments

[3] [5] [6] [7] [8] [9] such that the sum of all the data segments is equivalent to the value of the data block of that party. In proposed protocol quantity of data segments in a data block is kept equal to the number of parties [5]. The values of the segments are randomly selected by the party and it a secret of the party. If k be the number of segments (which is equal to the number of parties involved in the bus architecture) then in this protocol each party holds any one segment with it and k-1 data segments are sent to k-1 parties, one to each of the parties. Thus at the end of this rearrangement each of the parties holds k data segments in which only one data segment belongs to the party and other data segments belong to rest of parties presents in the network. In this proposed protocol, one of the parties is generally selected as the protocol initiator party which starts the computation by sending the data segment to the next party in the bus network. The receiving party adds its data segment and its secreted number and send to the next party presents in the architecture.

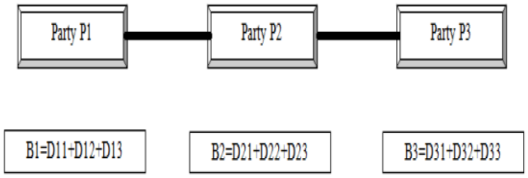

Figure 3 Rk secure sum process

This process is repeated until all the data segments of all the parties are added as well as data segments then the protocol initiator party is reduce the sum of all data segments then the sum is announced by the protocol initiator party. Now even if two adjacent parties maliciously cooperate to know the data of a middle party they will be able to know only those k data segments of a party which belong to every party. The sum of these data segments is a garbage value and thus worthless for the unauthorized parties. B1, B2 and B3 is a block of data then the segmentation break the block of data into the different number of data segments (D).

2.4.5 Modified Rk Secure Sum Protocol

When the concept come of distributed database [8] [12] [13] in which the whole database is divided into the number of parties and each party want that their own result will not known by the other parties so concept of security and privacy play a important role. In this paper we proposed modified Rk Secure Sum protocol for providing the highest privacy to the distributed database. In this proposed protocol all the parties are arranged in a sequential manner and party P1 is consider as a protocol initiator party. If there are N numbers of parties then number of round is also N. But the condition is that party P1 will always changes their position in each round till the party Pn. And after that P1 will disclose the result. First the P1 calculate their own partial support and added their own random number and send to the next party till Pn. After the completion of nth round party P1 will disclose the global result that accepted by all the parties presents in the distributed database. Algorithm shows formal working steps of modified Rk secure sum protocol.

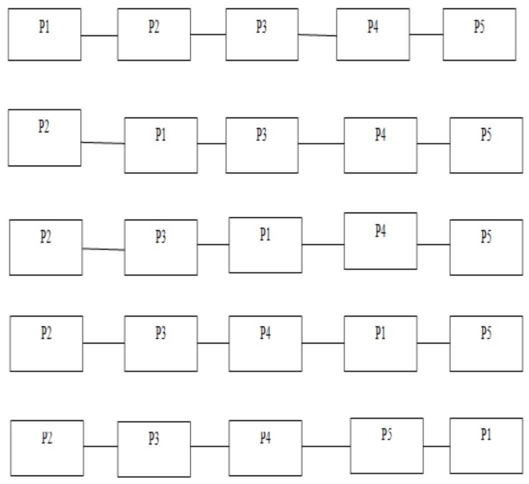

Figure 4 Modified Rk Secure Sum Process

Conclusion

This paper addresses the problem of computing global support within a scenario of homogeneous database. We assume that all sites have the same representation, but each site does not have information on different entities. The goal is to produce association rules that hold its input globally while limiting the information shared about each site. Many proposals have been cited to implement SMC. SMC being used in large scale databases which extends to preserve privacy to the private data of different sites. In this paper our focus is based on horizontal partitioned distributed data through a popular association rule mining technique.

References

[1]Agrawal, R., et al "Mining association rules between sets of items in large database". In: Proc. of ACM SIGMOD'93, D.C, ACM Press, Washington, pp.207-216, 1993.

[2]. Agarwal, R., Imielinski, T., Swamy, A. "Mining Association Rules between Sets of Items in Large Databases", Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-210, 1993.

[3]. Srikant, R., Agrawal, R "Mining generalized association rules", In: VLDB'95, pp.479-488, 1994.

[4]Agrawal, R., Srikant, R, "Privacy-Preserving Data Mining", In: proceedings of the 2000 ACM SIGMOD on management of data, pp. 439-450, 2000.

[5] Lindell, Y., Pinkas, B, "Privacy preserving Data Mining", In: Proceedings of 20th Annual International Cryptology Conference (CRYPTO), 2000.

[6]Kantarcioglu, M., Clifton, C, "Privacy-Preserving distributed mining of association rules on horizontally partitioned data", In IEEE Transactions on Knowledge and Data Engineering Journal, IEEE Press, Vol 16(9), pp.1026-1037, 2004.

[7] Han, J. Kamber, M, "Data Mining Concepts and Techniques". Morgan Kaufmann, San Francisco, 2006.

[8]Sheikh, R., Kumar, B., Mishra, D, K, "A Distributed k- Secure Sum Protocol for Secure Multi-Site Computations". Journal of Computing, Vol 2, pp.239-243, 2010.

[9]Sugumar, Jayakumar, R., Rengarajan, C "Design a Secure Multi Site Computation System for Privacy Preserving Data Mining". International Journal of Computer Science and Telecommunications, Vol 3, pp.101-105. 2012.

[10] N V Muthu Lakshmi, Dr. K Sandhya Rani ,"Privacy Preserving Association Rule Mining without Trusted Site for Horizontal Partitioned database", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, pp.17-29, 2012.

[11] N V Muthu lakshmi, Dr. K Sandhya Rani, "Privacy Preserving Association Rule Mining in

Horizontally Partitioned Databases Using Cryptography Techniques”, International Journal of Computer Science and Information Technologies(IJCSIT), Vol. 3 (1) , PP. 3176 – 3182, 2012.

[12] Goldreich, O., Micali, S. & Wigerson, A. ,”How to play any mental game”, In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, pp.218-229, 1987.

[13] Franklin, M., Galil, Z. & Yung, M.,”An overview of Secured Distributed Computing”. Technical Report CUCS- 00892, Department of Computer Science, Columbia University.