PRIVACY PRESERVING ASSOCIATION RULE MINING WITHOUT TRUSTED PARTY FOR HORIZONTALLY PARTITIONED DATABASES

N V Muthu Lakshmi[1] and Dr. K Sandhya Rani [2]

[1] Research Scholar: Dept of Computer Science, S.P.M.V.V, Tirupati, Andhra Pradesh, INDIA
nvmuthulakshmi@yahoo.co.in
[2] Professor: Dept of Computer Science S.P.M.V.V, Tirupati, Andhra Pradesh, INDIA
sandhyaranikasireddy@yahoo.co.in

ABSTRACT

Many data mining techniques are available to explore useful hidden information from large databases. Among these, association rule mining has wide applications to discover interesting relationships among attributes. The issue of privacy arises when the data is distributed among multiple sites and no site owner wishes to provide their private data to other sites but they are interested to know the global results obtained from mining process. In this paper a new model is proposed which utilizes hash based secure sum cryptography technique when no site can be treated as trusted party to find global association rules for horizontally partitioned databases.

KEYWORDS

Privacy Preserving Association Rule, Horizontally Partitioned Database, Cryptography Technique, Hash Based Secure Sum

1. INTRODUCTION

Due to the increased demand for knowledge discovery in all industrial domains, it is necessary to store all the raw data and to provide useful patterns with respective to the user needs. Generally, the storage of all raw data will be done in a database maintained by concerned organizations. Data mining techniques are available to retrieve useful information from large database. Prediction and description are the two fundamental goals of data mining. To full fill these goals many data mining techniques exists such as association rules, classification, clustering and so on. Among these, association rule has wide applications to discover interesting relationship among attributes in large databases. Association rule mining is used to find the rules which satisfy the user specified minimum support and minimum confidence. In the process of finding association rules, the set of frequent item sets are computed as the first step and then association rules are generated based on these frequent item sets.

Two types of database environments exist namely centralized and distributed. In contrast to the centralized data base model, the distributed data base model assumes that the data base is partitioned into disjoint fragments and each fragment is assigned to one site. The issue of privacy arises when the data is distributed among multiple sites and no site owner wish to

DOI : 10.5121/ijdkp.2012.2202

provide their private data to other sites but they are interested to know the global results obtained from the mining process.

Keeping in view of the motivation to incorporate privacy in data mining techniques to protect the confidential data of the user, there evolved a new stream in data mining era that is privacy preserving in data mining. There exists a key difference between regular data mining algorithms under various data mining techniques like classification, association, clustering and privacy preserving data mining algorithms that is the formal algorithms deals with how to analyze the stored raw data and how to extract useful knowledge discovery patterns from the database whereas in the later one, it mainly deals with the sensitive information of the user records where privacy factor is the major concern and it is considered to be vital issue.

The main aim in many distributed methods for privacy preserving data mining is to allow useful aggregate computations on the complete data set by preserving the privacy of the individual sites data/information. Each site owner is interested to collaborate in obtaining combined results, but not fully trust other sites in terms of the distribution of their own data sets. Any data mining system should satisfy the important property that is privacy preserving of data/information. Especially in distributed data mining, privacy preserving is one crucial aspect. Secure multi party computation is a useful approach to preserve the privacy in distributed data mining. Privacy preserving data mining utilizes a mining algorithm to procure mutually beneficial global data mining objectives without revealing private data. Therefore, in many data mining applications privacy preserving has become an important issue.

2. PRIVACY PRESERVING ASSOCIATION RULE MINING

In privacy preserving distributed data mining, how the data is partitioned among different sites is very important. The three main partitioning methods in distributed data base environment are horizontal, vertical and mixed mode. In case of horizontal partition, the same schema is used to keep the data at each site whereas in vertical partition, different schemas are used at different sites, that is, different kinds of data on the same entities. The other partitioning method is mixed partitioning where data is partitioned horizontally and then each fragment is further partitioned into vertical and vice versa. Privacy preserving association rule mining algorithms can be divided into three categories according to privacy protection technologies. The three categories are heuristic-based techniques, reconstruction-based techniques and cryptography-based techniques. In this paper cryptographic approach is adopted to find global association rules by preserving the privacy when no party can be treated as trusted party. The cryptography approach is very popular for the following two reasons:

- It has a well established and well defined model meant for privacy which can actually provide good number of methodologies for verifying and validating purpose.
- Cryptography branch has a wide variety of tool set to incorporate privacy in data mining.

Some of the relevant works in privacy preserving data mining are presented as follows:

An overview of data mining techniques and a detailed description of mining association rules are presented by the authors and this survey is done keeping in view of data base researcher's point of view based upon the data mining techniques. The authors also discussed various classes of data mining techniques and its contrasting features exists among them [1].Secure two party computations concept was first introduced by Yao [2], and later generalized to multi party computation. In [3], the authors presented ID3 classification for two parties with horizontally partitioned data by using secure protocols to achieve complete zero knowledge leakage. The authors proposed in [4], four efficient methods namely secure sum, secure set union, secure size of set intersection and scalar product for privacy preserving data mining in distributed environment. In [5], the authors discussed the problem of privacy preserving data mining of association rules when the data is partitioned horizontally. They proposed algorithm which uses three basic ideas such as randomization, encryption of site results and secure computation. The state of art in the area of privacy preserving data mining techniques is presented [6]. The authors also discussed about classifications of privacy preserving techniques and privacy preserving algorithms such as heuristic-based techniques, cryptography-based techniques, and reconstruction based technique. A framework for evaluating privacy preserving data mining algorithms and based on this frame work one can assess the different features of privacy preserving algorithms according to different evaluation criteria [7]. An enhanced kantarcioglu and Clifton's schemes is proposed by authors in [8], which is a two phase for privacy preserving distributed data mining. In [9], the authors discussed the problem of privacy preserving data mining in distributed data bases. They suggested a new paradigm based on two separate entities, a minor and a calculator, both are not having any parts of the data base. They also presented three algorithms based on this paradigm, one for horizontally partitioned data, one for vertically partitioned data and one for any data mining method. The authors in [10] proposed a new algorithm for mining association rules in distributed homogeneous databases based on semi-honest model and negligible collision probability. In [11], the authors presented a classification, an extended description and clustering of various association rule mining algorithms. They also suggested further research directions of privacy preserving association rule mining algorithms by analyzing the existing work.

2.1. Distributed Association Rule Mining

Among many data mining techniques, association rule mining is receiving great attention from researchers due to its usage in many applications. An association rule can be defined formally as follows:

Let I = $\{i_1, i_2, \ldots, i_m\}$ be the set of attributes called items. The item set X consisting of one or more items. Let DB = $\{t_1, t_2, \ldots, t_n\}$ be the database consisting of n number of boolean transactions, and each transaction t_i consisting of items supported by i^{th} transaction. An item set X is said to be frequent when number of transactions supporting this item set is greater than or equal to the user specified minimum support threshold otherwise it is said to be infrequent. An association rule is an implication of the form X→ Y where X and Y are disjoint subsets of I, X is called the antecedent and Y is called consequent. An association rule X→Y is said to be strong association rule only when its confidence is greater than or equal to user specified minimum confidence.

As many large data bases are distributed in nature, association rule mining requires substantial processing power to operate on a distributed system.

The procedure to declare an item set X as globally frequent or not in horizontally partitioned databases is explained below:

Let $Site_1$,$Site_2$, …,$Site_n$ are n sites in a distributed system and the global database DB is horizontally divided into DB_1,DB_2, …, DB_n non overlapping n partitions such that DB $= \sum_{i=1}^{n} DB_i$ .The partition DB_i is assigned to $Site_i$ .The length of DB that is $|DB| = |DB_1| + |DB_2| +\ldots+ |DB_n|$.

To determine the global support of an item set, each site's local supports are required.
The local support count of an item set X at site $Site_i$ is X.sup for $1 \leq i \leq n$. The item set X is locally frequent at $Site_i$ if X.sup $\geq$ MinSup * $|DB_i|$. The global support count of an item set X is computed as

$$\text{Global support of item set, X} = \sum_{i=1}^{n} X.\text{sup}$$

If global support of X is greater than or equal to MinSup *|DB| (global threshold) then X is globally frequent otherwise X is globally infrequent item set.
For finding privacy preserving association rule mining, a new model is proposed and presented in the next section.

3. PROPOSED MODEL

In distributed environment, the method of finding privacy preserving association rules for horizontally partitioned distributed databases of n sites (n >2), where no site is considered to be trusted party is proposed in this paper. Each site wish to find global results which are computed from all sites database that is global frequent item sets with support values by involving itself in the mining process in indirect form by providing local frequent item sets to its successor sites in disguised form.

Every site in the distributed environment requires global association rules which provide useful information to analyze their problems easily and to improve the performance of their activities or services. But global association rules can be generated when they have global frequent item sets and their supports. The global frequent item sets are determined from a set of local frequent item sets which are generated by the sites. An item set which is locally frequent in one or more sites need not be frequent at globally and an item set which is infrequent in one or more sites need not be infrequent at globally since sum of the item set's support value must be higher than minimum number of transactions required to support globally. The challenging task here is how to find whether an item set which is locally frequent in at least one site is globally frequent or globally infrequent using sum of support values of sites where no single site is willing to provide their supports since any site can get benefits if he knows the support values of other sites. As the support value decides whether an item set is frequent or not, every site treats support values of their item sets as private information which are to be protected from others.

In the proposed model, cryptography technique is used by adopting a new concept called hash based secure sum technique to find global frequent item sets and their global supports. This new technique extracts local support value of each individual site in disguised form however no site can predict the support of any item set of any site. This hash based secure sum cryptography technique can be used easily and efficiently to find global frequent item sets and their supports for n number of sites in distributed environment and these global results are used to generate global association rules based on user specified minimum confidence threshold. The following diagram shows the communication among three sites.

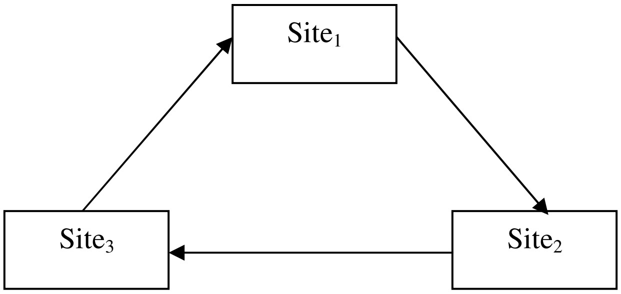

Figure 1: Communication among three sites

In this model, communication between sites will be performed in cyclic order and every site has successor and predecessor and the last site's successor is the first site. The successor site of i^{th} site can be computed from (i mod n) + 1.

Every site receives information from its predecessor and sends information to its successor site only. This model does not have data transfers from a site to its predecessor site. Among n sites, the first site in the order is called as initiator site which initiates the mining process to find global frequent item sets.

Only the first site, $Site_1$ knows the total database size that is total number of transactions in all sites but the $Site_1$ do not know any site's database size. The $Site_1$ is also just like other site, consists of a database and participates in the mining process. This site has few privileges, to know the minimum support threshold value and to find actual support of any item set which is globally frequent. The $Site_1$ broadcast the minimum support threshold value as specified by the user to all the other sites. Finally $Site_1$ finds the actual support of an item set which is globally frequent based on the accumulated excess support values computed at all sites.

This model has two phases to find globally frequent item sets by preserving the privacy of the individual's private data in a situation, where no site can be considered as trusted site.

Phase I

Step1: Every site owner applies frequent item set generation algorithm for their database based on minimum support threshold which has been received from $Site_{1.}$

Step2: Each site should maintain both frequent item set list and infrequent item sets list with support values separately.

Step3: Each site generates unique random number and sends this value to its successor site. So every site will have two random numbers, one is its own and another one received from predecessor site.

The reason for maintaining infrequent item set list with support values in step 2 is as follows: The infrequent item set in a site may be frequent in any other site/sites and by summation of local supports; the same item set may become globally frequent. Because of this reason, an infrequent item set list at each site is needed for the computation of partial support of an item set which may be frequent in any other site/sites.

Phase II

The various steps in Phase II are as follows:

Step1: The first site, Site$_1$ prepares a list which consists of all local frequent items of its database and 5% of infrequent items whose support value is nearer to minimum support (positive border item set). Partial support values are computed for each item set in this list by using the following formula

$PS_j = X_j.sup - MinSup * |DB_1| + RN_1 - RN_n$ where RN_n is the random number received from nth site,RN_1 is its own random number.

Site$_1$ computes a mask value by using double hash function and added to the PS_j values to get the value in disguised form as

$PS_j = \quad PS_j + MaskValue$

Finally the first site prepares a list which consists of item sets and their PS_j values and sends to its successor site, Site$_2$.

Step2: Site$_2$ computes PS_j for each item set in the received list using the following formula

$PS_j = PS_j + X_j.sup - MinSup * |DB_i| + RN_i - RN_{i-1}$

where PS_j value on the RHS of the above formula indicates partial supports received from predecessor site Site$_1$.

The computed partial support values for all item sets at Site$_2$ will be sent to the successor site Site$_3$.

Step3: Every site in the sequence of sites, Site$_3$, Site$_4$, …, Site$_n$, performs step2 by using its respective values and sends its computed results to its successor.

Step4: At the end, Site$_n$ computes PS_j values for each item set in the received list from its predecessor site and sends to Site$_1$. These PS_j values are nothing but cumulated partial support of n sites.

Step5: Then Site$_1$ which is the initiator of the process finds the excess support from these disguised form values by subtracting mask value from partial supports which are received from nth site.

Step6: Site$_1$ finds whether an item set is globally frequent or not by comparing excess support of an item set with zero. If this value ≥ 0 then the item set is declared as globally frequent otherwise globally infrequent.

Step7: Finally Site$_1$ prepares a list consisting of global frequent item set with their supports and sends to the successor site, Site$_2$. Also request to take the initiation to perform the above steps for its local frequent item sets which are not yet processed.

Step8: Site$_2$ perform the above steps 1 to 6 to find the global frequent item sets for its database. These global frequent item sets with their supports are placed in a list and then appends to a received list from predecessor site which consists of global frequent item sets and supports declared by the predecessor site. Site$_2$ sends this list to its successor site and request to initiate the process.

Step9: Every site in the sequence of remaining sites, Site$_3$, Site$_4$, …, Site$_n$, performs steps 7 & 8 by using its respective values until no more local frequent item sets in any site.

Step10: Finally Site$_n$ sends the appended globally frequent item set list with excess support values to Site$_1$. As Site$_1$ knows the total database size, it computes the actual support for each global frequent item set in the list by using the following formula

$$AS_j = PS_j + MinSup * |DB|$$

The PS_j in the above formula possess cumulative value of partial support of item set X_j in all sites

Step11: $Site_1$ broadcast the final list of global frequent item set with actual support values to all sites.

Step12: Each site can generate association rules with various confidence values for each global frequent item set in the final list received from $Site_1$.

The calculation of mask value in Step1 is as follows:
Mask value is computed by applying two different hash functions (double hashing) one after another.The initiator site generates a key and which is the input to the first hash function.
$Key_1 = Hash_1(Key) = key \bmod M$ where M is an integer lies between $2 \leq M \leq 10$.
The second hashing function is applied by taking Key_1 as input and it returns a value called MaskValue.

$$MaskValue = Hash_2 (Key_1) = Key \pm M^{Key1}$$

In the above formula plus (+) operator is used when the value of key1 is even otherwise minus (-) operator is used.
The double hashing function which is used to find the mask value enhances the privacy by making partial supports in more disguised form.

When a site takes initiation for finding global frequent item sets for its local frequent item sets, the site also includes few infrequent item sets to the list of frequent item sets. This makes confusion to the successor site in knowing the local frequent item sets of predecessor site and thus protects the local frequent item sets of a site from the remaining sites.

4. IMPLEMENTATION OF PROPOSED MODEL WITH SAMPLE DATABASES

The proposed model is illustrated by using three horizontally partitioned distributed databases for finding privacy preserving association rule mining when no party is considered as trusted. There are three sites exist termed as $Site_1$, $Site_2$ and $Site_3$ and possess different databases DB_1, DB_2, DB_3 respectively. All three site's database having common attributes but different transactions and every site has local autonomy over its database. Databases at three sites are given in the following tables.

TID\Item	X_1	X_2	X_3	X_4	X_5
$Site_1$ has the following database					
T_1	1	0	0	1	0
T_2	1	1	0	0	1
T_3	0	1	1	1	1
T_4	0	0	1	0	0
T_5	1	0	0	1	0
T_6	0	0	1	1	0

T_7	1	0	1	0	1

Table 1. Database, DB$_1$ at Site$_1$

TID\Item	X_1	X_2	X_3	X_4	X_5
Site$_3$ has the following database					
T_1	1	1	0	1	0
T_2	1	0	0	1	0
T_3	1	1	1	0	1
T_4	0	0	1	0	1
T_5	1	1	1	1	0
T_6	0	1	1	0	0
T_7	1	0	0	1	1

Table 2. Database, DB$_2$ at Site$_2$

TID\Item	X_1	X_2	X_3	X_4	X_5
Site$_2$ has the following database					
T_1	1	0	0	1	0
T_2	1	1	1	1	1
T_3	0	1	1	0	1
T_4	0	0	1	1	1
T_5	1	1	0	1	1
T_6	0	1	1	1	0

Table3. Database DB$_3$ at Site$_3$

Let us assume that the user specified minimum support threshold value is 40%. Minimum number of transactions required (MinNumTrans) to support any item set in any site can be determined from 40% of |DB$_i$| for i from 1 to 3. According to this threshold value, each site must have the following number of transactions required to support an item set to be frequent.

For Site$_{1,}$ MinNumTrans = 3, for Site$_2$, MinNumTrans = 2, for Site$_3$, MinNumTrans = 3.

Therefore an item set to be globally frequent, the minimum number of transactions required is 8.

In phase I, each site finds a list consisting of local frequent item sets with their supports using frequent item set generation algorithm based on MinSup threshold value 40%. Each site's local frequent item sets are shown below.

At Site$_1$:

The list of local frequent item sets are {X_1, X_3, X_4, ,X_5} and the remaining item sets are infrequent.

At Site$_2$:

The list of local frequent item sets are { X_1 , X_2, X_3, X_4, X_5 , (X_1 ,X_2), (X_1 ,X_4), (X_1 ,X_5), (X_2,X_3),(X_2,X_4), (X_2,X_5),(X_3,X_4),(X_3,X_5),(X_4,X_5),(X_1 ,X_2,X_4), (X_1 ,X_2,X_5), (X_2,X_3,X_4), (X_2,X_3,X_5), (X_2,X_4,X_5), (X_3,X_4,X_5), (X_1 ,X_4,X_5) (X_1 ,X_2,X_4,X_5)} and the remaining item sets are infrequent.

At Site$_3$:

The list of local frequent item sets are {X_1 , X_2, X_3, X_4, X_5, (X_1 ,X_2), (X_1 ,X_4),(X_2,X_3)} and the remaining item sets are infrequent.

Let us illustrate the computations involved in the proposed model in finding whether an item set is globally frequent or infrequent by taking two item sets as follows.

Consider the item sets {X_2, (X_1, X_2)} and denoted as I_1 = (X_2) and I_2 = (X_1, X_2)

Random numbers generated by Site$_1$, Site$_2$ and Site$_3$ are RN_1 = 15, RN_2 = 110 and RN_3 = 56 respectively.

Consider the item set I_1.

I_1= (X_2).

Site$_1$, initiates the process to find global results by generating key value 225 and M value 4.

Even though the item set I_1 is infrequent at Site$_1$, it is added to a frequent item set list as dummy item set.

Site$_1$ has the following values

RN_1 = 15, RN_3 =56, key = 225 and M = 4

Key1 = Key mod M

=225 mod 4 = 1

Maskkey = Key - M ^{key1} = 225 - 4^1 =221

Site$_1$ computes partial support value as

PS_1 = (I_1).sup –MinSup * |DB_1| + (RN_1- RN_3) + Maskkey

= 2 – 3 – 41 + 221 = - 42 + 221 = 179

Now $Site_1$ sends this PS_1 to its successor site that is $Site_{2.}$

The $Site_2$ computes partial support value as follows

$PS_1 = (I_1).sup - MinSup * |DB_2| + (RN_2 - RN_1) + PS_1 = 4 – 2 + 95 + 179 = 276$

Now sends this to $Site_3$ and then $Site_3$ computes partial support as

$PS_1 = (I_1).sup - MinSup * |DB_3| + (RN_3 - RN_2) + PS_1 = 4 – 3 – 54 + 276 = 223$

Now sends this to $Site_1$.

Since $Site_1$ is the initiator for this item set, subtracts MaskKey value from received partial support value as

$PS_1 = 223 -221= 2$

Since PS_1 is greater than or equal to zero, $Site_1$ declares the item set (X_2) as globally frequent even it is infrequent at this site. The actual support value of (X_2) can be computed by adding 40% of |DB| which is 8 (minimum number of transactions required to support any item set to be globally frequent) to PS_1 value as follows:

Actual support of an item set (X_2) = 2 + 8 = 10

Hence the global support of a global frequent item set (X_2) is 10.

The item set (X_2) proves that an item set which is infrequent at $Site_1$ and frequent at $Site_2$ and $Site_3$ may become globally frequent after doing various computations in various sites. It also proves that adding a dummy item set that is infrequent to the frequent item set list at initiation site may become frequent at globally and helps to make confusion to the successor site not to predict the local frequent item sets of predecessor site.

Consider the second item set I_2, $I_2 = (X_1, X_2)$

The process is initiated by $Site_2$ as I_2 is infrequent at $Site_1$ and frequent at $Site_{2.}$

$Site_2$ (Inititator site) initiates the process by generating the key value as 350 and M value as 8.

$Site_2$ has the following values

$RN_2 = 110$, $RN_1= 15$, Key = 350, M = 6

Key_1 = Key mod 8 = 350 mod 6 = 2

Maskkey = Key - M^{key1} = 350 + 6^2 = 350 + 36 = 386
partial support can be computed as

$PS_2 = (I_2).sup - MinSup * |DB_2| + (RN_2 - RN_1) + Maskkey$

$= 2 – 2 + 95 + 386 = 481$

Send this to successor site that is $Site_{3.}$ The $Site_3$, computes partial support as

$PS_2 = (I_2).sup – MinSup * |DB_3| + (RN_3 - RN_2) + PS_2 = 3 – 3 - 54 + 481 = 427$

Then $Site_3$ sends this computed result to $Site_1$.

Now the $Site_1$ finds PS_2 as

$PS_1 = (I_2).sup – MinSup * |DB_1| + (RN_1 - RN_3) + PS_2 = 1 – 3 – 41 + 427 = 384$

Now sends this to $Site_2$

Since Site2 is the initiator for this item set, subtracts MaskKey from PS_2

= 384 - 386 = -2

Hence the item set (X_1,X_2) is globally declared as infrequent even though it is frequent at $Site_2$ and $Site_3$.

The item set $(X_{1,} X_2)$ proves that an item set which is frequent in some sites and infrequent in some other sites becomes globally infrequent.

The above process can be used to find all frequent item sets as globally frequent or not.

The procedure specified in section 3 is applied at all sites for the databases DB_1,DB_2 and DB_3 to find the global frequent item sets and the final results are shown in the following table.

Table 4. Global Frequent Item Sets and Their Supports

Item set	Sup	Item set	Sup	Item set	Sup
X_1	12	X_3	12	X_5	10
X_2	10	X_4	13	(X_1,X_4)	9

The global frequent item sets such as X_1, X_3, X_4, X_5 are locally frequent in all three sites where as the global item sets (X_2) and (X_1, X_4) are infrequent at $Site_1$ and frequent at $Site_2$ & $Site_3$ This clearly reveals that to declare an item set as globally frequent or not, requires the support of item set values at all sites.

5. PERFORMANCE OF THE PROPOSED MODEL

A new model which utilizes hash based secure sum cryptography is proposed in this paper for horizontally partitioned databases without trusted party to find global association rules. The efficiency of the proposed method in terms of privacy and communication is discussed as follows:

- In the process of computing partial support value of each item set at each site, MinSup * $|DB_i|$ is subtracted from its local support value and then a value is added which is computed by subtracting received predecessor 's random number from its own random number. So, finally the partial support value of an item set is obtained in disguised form.

To avoid the successor site from guessing the predecessor site's private data/information, a double hash function is defined in this paper and is used to find the mask value which will be added to the disguised form of partial support value to enhance the privacy further. As mask value is computed by applying two different hash functions which perform many arithmetic computations such as modulus, addition, subtraction and exponentiation, it is not possible for any successor site to predict predecessor site's data/information from the received partial support values.

- Each initiation site prepares a list which consists of all locally frequent item sets of its database and 5% of infrequent item sets (positive border item sets) whose support value is nearer to the minimum support to find whether the item sets in this list are globally frequent or not by extracting each other site's local supports. Few infrequent item sets are added to a list of frequent item sets to avoid the situation that a successor site can predict local frequent item sets of its predecessor site.
- Site_1 is aware of the total database size and has final accumulated partial supports of all frequent item sets of all sites. Based on this information, Site_1 can find actual support of all globally frequent item sets but it is impossible for Site_1 to find any site or sites local supports of any item set since it receives accumulated excess support values only.
- Finally every site obtains the list of global frequent item sets with support values received from Site_1. Based on this list no site can predict the contribution of other site's database which makes the item sets globally frequent as global frequent item sets may or may not be frequent in all sites.

The communication cost is measured in distributed environment based on the number of communications for data transfers among n sites. The number of data transfers at different stages in the proposed model is specified as follows:

- To broadcast minimum support threshold by Site_1 to n-1 number of sites requires (n-1) data transfers.
- n number of data transfers are required for sending each predecessor site's random number to its successor site.
- A site which initiates to find global frequent item sets from its local frequent item set list requires n number of data transfers. This task is to be performed at all remaining sites. Hence the total number of data transfers required to find all the global frequent item sets of n sites is n^2. This is the maximum number of data transfers required. However if any site or sites are not having local frequent item sets which are not processed so far, needs less number of data transfers compared to n^2 data transfers. The minimum number of data transfers required always depends on the databases of sites.
- The Site_1 requires (n-1) data transfers to broadcast global frequent item sets along with actual global support.

The tasks specified in the last two points perform bulk data transfers for several item sets instead of single data transfer for each item set.

From these discussions, the proposed model is efficient in finding privacy preserving association rule mining when no site can be treated as trusted party for horizontally partitioned databases.

6. CONCLUSION

The problem of preserving privacy in association rule mining when the database is distributed horizontally among n (n>2) number of sites when no trusted party is considered. A model which adopts a hash based secure sum cryptography technique to find the global association rules is proposed in this paper by preserving the privacy constraints. Double hashing function is adopted to enhance the privacy further. The proposed model efficiently finds global frequent item sets even when no site can be treated as trusted. By taking sample databases, working of the proposed model is explained. Efficiency of the proposed model is analyzed in terms of privacy and communications and it shows that the proposed model easily and efficiently finds the global frequent item sets by satisfying all the privacy constraints. This model can be applied for any number of sites and for any number of transactions in the databases of sites.

REFERENCES

[1] Ming-Syan Chen, Jiawei Han,Yu, P.S. (1996), Data mining: an overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 8 No. 6, pp 866 – 883.

[2] A.C Yao(1986), How to generate and exchange secrets, In proceedings of the 27th IEEE Symposium on Foundations of Computer Science, pp 162-167.

[3] Y Lindell and B pinkas (2000), Privacy preserving data mining, In Proc. O CRYPTO'00, pp36-54. Springer-Verlag2000.

[4] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu(2003), Tools for privacy preserving distributed data mining, SIGKDD Explorations, Vol. 4, No. 2 pp 1-7.

[5] M. Kantarcioglu and C. Clifto (2004). Privacy-preserving distributed mining of association rules on horizontally partitioned data. In IEEE Transactions on Knowledge and Data Engineering Journal, volume 16(9), pp. 1026-1037.

[6] Verykios, V.S., Bertino, E., Nai Fovino, I., Parasiliti, L., Saygin, Y., and Theodoridis, Y. (2004), State-of- the-art in privacy preserving data mining, SIGMOD Record, 33(1):50–57.

[7] Elisa Bertino , Igor Nai Fovino Loredana Parasiliti Provenza (2005), A Framework for Evaluating Privacy Preserving Data Mining Algorithms, Data Mining and Knowledge Discovery, Vol. 11, 121–154.

[8] Chin-Chen Chang, Jieh-Shan Yeh, and Yu-Chiang Li (2006), Privacy-Preserving Mining of Association Rules on DistributedDatabases, IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.11.

[9] Alex Gurevich, Ehud Gudes (2006), Privacy preserving data mining algorithms without the use of secure computation or perturbation, 10th international database Engineering and Applications Symposium IDEAS06 IEEE.

[10] Mahmoud Hussein, Ashraf El-Sisi,and Nabil Ismail (2008), Fast Cryptographic Privacy Preserving Association Rules Mining on Distributed Homogenous Data Base, I. Lovrek, R.J. Howlett, and L.C. Jain (Eds.): KES 2008,Part II,LNAI 5178, pp. 607–616, 2008.© Springer-Verlag Berlin Heidelberg.

[11] Jian Wang, Yongcheng Luo, Yan Zhao, Jiajin Le(2009), A Survey on Privacy Preserving Data Mining, First International Workshop on Database Technology and Applications, pp. 111-114.