ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Secure Mining of Association Rules in Horizontally Distributed Databases

Sonal Patil[1], Harshad Patil[2]

[1]Assistant Professor, Department of Computer Science Engineering, GHRIEM, Jalgaon.
[2]PG Scholar, Department of Computer Science Engineering, GHRIEM, Jalgaon.

Abstract: We propose a protocol for secure mining of association rules in horizontally distributed databases. Our protocol is optimized than the Fast Distributed Mining (FDM) algorithm which is an unsecured distributed version of the Apriori algorithm. The main purpose of our protocol is to remove the problem of mining generalized association rules that affects the existing system. Our protocol offers more enhanced privacy with respect to previous protocols. In addition, it is simpler and is optimized in terms of communication rounds, communication cost and computational cost than other protocols .

Index Terms: Data Mining, Databases, Apriori Algorithm, FDM, Association Rule Mining.

I. INTRODUCTION

Data mining is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable pattern in data. Data mining represents the integration of several fields, including machine learning, database systems, data visualization, statistics and information theory. Several major data mining techniques have been developed and used in data mining projects recently including association, classification, clustering, prediction and sequential patterns [1].

Knowledge discovery in databases is a complex process, which covers many interrelated steps. Key steps in the knowledge discovery process are-

DATA SELECTION- The data needed for the data mining process may be obtained from many different and heterogeneous data sources. This first step obtains the data from various databases, files and non-electronic sources.

DATA PREPROCESSING- The data to be used by the process may have incorrect or missing data. There may be anomalous data from multiple sources involving different data types and metrics. There may be many different activities performed at this time. Erroneous data may be corrected or removed, whereas missing data must be supplied or predicted.

DATA TRANSFORMATION- Data from different sources must be converted into common format for processing. Some data may be encoded or transformed into more usable formats. Data reduction may be used to reduce the number of possible data values being considered.

DATA MINING- This step applies algorithms to the transformed data to generate the desired results.

DATA PATTERN EVALUATION/ INTERPRETATION- How the data mining results are presented to the users is extremely important because the usefulness of the results is dependent on it. Various and GUI strategies are used at this last step.

Knowledge Discovery in Databases (KDD) is an automated extraction of novel, understandable and potentially useful patterns implicitly stored in large databases, data warehouse and other massive information repositories. KDD is a multi-disciplinary field drawing work from areas including database technology, artificial intelligence, machine learning, neural networks, statistics, pattern recognition, information retrieval, high performance computing and data visualization [1].

Privacy preserving data mining [2, 3] is a new investigation in data mining and statistical databases [4]. In PPDM data mining algorithms are analyzed for side effects obtain in data privacy. Two fold consideration in privacy preserving data mining. First is sensitive raw data that are kept secure from unauthorized access like identifiers, names ,addresses should be modified from original database in order for receiver of data not to be able to compromise another person's privacy. Second is sensitive knowledge is excluded that can be mined from a database by using data mining algorithms as such type of knowledge compromises data privacy[4].

II. RELATED WORK

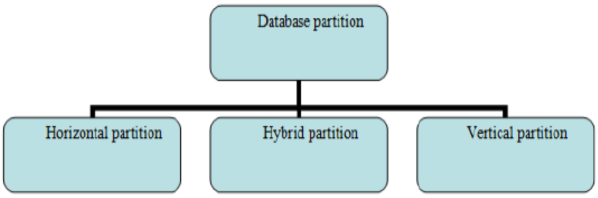
Partitioning of Database: - Data can be partitioned in three different ways that is, like horizontally partitioned data, vertically partitioned data or mixed partitioned data.

Horizontal partitioning: - The data can be partitioned horizontally where each fragment consists of a subset of the records of relation R. Horizontal partitioning [3] [9] [10] [11] divides a table into several tables. The tables have been partitioned in such a way that query references are done by using least number of tables else excessive UNION queries are used to merge the tables sensibly at query time that can affect the performance.

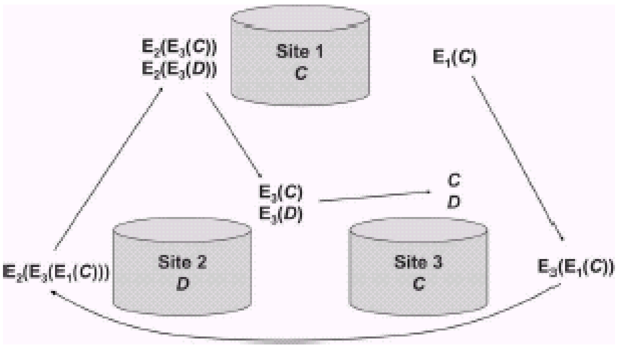
Vertical partitioning: - The data can be divided into a set of small physical files each having the subset of the original relation, the relation is the database transaction that normally requires the subsets of the attributes.

Mixed partitioning: - The data is first partitioned horizontally and each partitioned fragment is further partitioned into vertical fragments and vice versa.

The idea is to build up a well organized method that enables a secure computation along with minimizing the amount of private data that each party discloses to other. Privacy preserving association rule mining may be used to solve these problems for horizontally partitioned database.



There is a description of the literature relevant to our field and topic, for the purpose of survey to understand the domain of problem and possible approaches to solve the problem and improvements suggested by different authors so that it is possible to design the new algorithm which may solve the problems with some improvements. In this section, the summary of different algorithm, problems associated with them and the different approaches used by different authors to solve those problems are discussed. As suggested by V.S. Verykios, A.K. Elmagarmid, B. Elisa, Y. Saygin, and D. Elena [7], there is large depository of data that contains sensitive data that must be protected from illegitimate access. As we know that protection of data from illegal access is a long term goal for database security research community, governmental organizations and business associations. As advancement in data mining arises with that risk of releasing data to outside party also increases. Every disclosure method affects data in some way and modifies to true value and relationship. In this paper, they investigate confidentiality of a broad category of association rules. In this they presented algorithms and strategies for hiding a group of association rules is characterized as sensitive. A rule is sensitive if its disclosure risk is above certain privacy threshold. They perform an evaluation study of the hiding algorithms in order to analyze their time complexity and the impact that they have in the original database. The security impact of data mining is analyzed and some possible approaches to the problem of inference and discovery of sensitive knowledge in a data mining context are suggested. The proposed strategies include fuzzyfying and augmenting the source database and also limiting the access to the source database by releasing only samples of the original data. Clifton adopts an approach in that he studied the correlation between the amount of released data and the significance of the patterns that were discovered. They also show how to determine the sample size in such a way that data mining tools cannot obtain reliable results. Clifton and Marks is also recognize the necessity of analyzing the various data mining algorithms in order to increase the efficiency of any adopted strategy that deals with disclosure limitation of sensitive data and knowledge. The solution proposed by Clifton in is independent from any specific data mining technique; other researchers propose solutions that prevent disclosure of confidential information for specific data mining algorithms such as association rule mining and classification rule mining. Classification mining algorithms may use sensitive data to rank objects; each group of objects has a description given by a combination of non sensitive attributes. The sets of descriptions, obtained for a certain value of the sensitive attribute, are referred to as description space. For Decision-Region-based algorithms, the description space generated by each value of the sensitive attribute can be determined apriori. The authors in first identify two major criteria which can be used to assess the output of a classification inference system and then they use these criteria, in the context of Decision-Region based algorithms, to inspect and to modify, if necessary, the description of a sensitive object so that they can be sure that it is not sensitive. In this research they presented two fundamental approaches in order to protect sensitive rules from disclosure. The first approach prevents rules from being generated by hiding the frequent sets from which they were derived. The second approach reduces the importance of the rules by setting their confidence below a user-specified threshold. In this research they presented two fundamental approaches in order to protect sensitive rules from disclosure.



As suggested by Kazem Taghva, Pavankumar Bondugula, Darshana Gala [6]. An association rule expresses the dependency of a one set of attributes on another attributes. In the identification of association rules, an Apriori algorithm is one of the known techniques that is used. In this research, this technique is used for privacy data identification and extraction from printed documents. In this research they point the problem of discovering association rules for various privacy types from printed documents. An association rule expresses the dependence of a set of attribute-value pairs and upon another set of items (itemset). The mining of association rules is performed in two stages: The frequent sets of items from the data discovery and association rules generation from the frequent item sets. Searching of these frequent itemsets is in general a combinatorial expensive task. Association rule mining has a broad range of applicability. It was first introduced to find the association between items in supermarket transactions for promotion of sales, arrangement of associated items accordingly, to increase profits etc.

Private Association Rule Mining Overview:

Our method follows the two-phase approach described above, but combining locally generated rules and support counts is done by passing encrypted values between sites.

The two phases are discovering candidate itemsets (those that are frequent on one or more sites) and determining which of the candidate itemsets meet the global support/confidence thresholds.The first phase (Fig. 1) uses commutative encryption. Each party encrypts its own

frequent itemsets (e.g., Site 1 encrypts itemset C). The encrypted itemsets are then passed to other parties until all parties have encrypted all itemsets. These are passed to a common party to eliminate duplicates and to begin decryption. (In the figure, the full set of itemsets are shown to the left of Site 1, after Site 1 decrypts.) This set is then passed to each party and each party decrypts each itemset.

The final result is the common itemsets (C and D in the figure).In the second phase (Fig. 2), each of the locally supported itemsets is tested to see if it is supported globally.

In the figure, the itemset ABC is known to be supported at one or more sites and each computes their local support. The first site chooses a random value R and adds to R the amount by which its support for ABC exceeds the minimum support threshold. This value is passed to site 2, which adds the amount by which its support exceeds the threshold (note that this may be negative, as shown in the figure.) This is passed to site 3, which again adds its excess support. The resulting value (18) is tested using a secure comparison to see if it exceeds the Random value (17). If so, itemset ABC is supported globally.

There are several fields where related work is occurring. We first describe other work in privacy-preserving data mining, then go into detail on specific background work on which this paper builds. Previous work in privacy-preserving data mining has addressed two issues. In one, the aim is preserving customer privacy by distorting the data values [2]. The idea is that the distorted data does not reveal private information and thus is sa fe‖ to use for mining. More recently, the data distortion approach has been applied to Boolean association rules. Again, the idea is to modify data values such that reconstruction of the values for any individual transaction is difficult, but the rules learned on the distorted data are still valid. One interesting feature of this work is a flexible definition of privacy, e.g., the ability to correctly guess a value of ―1‖ from the distorted data can be considered a greater threat to privacy than correctly learning a 0 .‖The other approach uses cryptographic tools to build decision trees. In this work, the goal is to securely build an ID3 decision tree where the training set is distributed between two parties. The basic idea is that finding the attribute that maximizes information gain is equivalent to finding the attribute that minimizes the conditional entropy. The conditional entropy for an attribute for two parties can be written as a sum of the expression of the form (v1+ v2)*log(v1+v2). The authors give a way to securely calculate the expression (v1+v2)*log(v1+v2) and show how to use this function for building the ID3 securely.

1) Mining of Association Rules:

The association rules mining problem can be defined as follows [1]: Let I ={ i1,i2, . . .,in} be a set of items. Let DB be a set of transactions where each transaction T is an itemset such that T I. Given an itemset X I, a transaction T contains X if and only if X T. An association rule is an implication of the form X =>Y, where X I, Y I, and X Y= . The rule X =>Y has support s in the transaction database DB if s% of transactions in DB contains X Y. The association rule holds in the transaction database DB with confidence c if c% of transactions in DB that contain X also contains Y. An itemset X with k items is called kitemset.

The problem of mining association rules is to find all rules whose support and confidence are higher than certain user-specified minimum support and confidence. In this simplified definition of the association rules, missing items, negatives, and quantities are not considered. In this respect, transaction database DB can be seen as 0/1 matrix where each column is an item and each row is a transaction.

In this paper, we use this view of association rules.

2) Distributed Mining of Association Rules:

The above problem of mining association rules can be extended to distributed environments. Let us assume that a transaction database DB is horizontally partitioned among n sites (namely, S1,S2, . . . , Sn) where DB = DB1 DB2 . . . DBn and DBi resides at side Si(1<=I<=n).

The itemset X has local support count of X.supi at site SI if X.supI of the transactions contains X. The global support count of X is given as X.sup = N

I=1 X.supi. An itemset X is globally supported if X.sup>=s* N

I=1 DBi . Global confidence of a rule X =>Y can be given as {X Y}.sup/X.sup.A fast algorithm for distributed association rule mining is given in Cheung et al. [1]. Their procedure for fast distributed mining of association rules (FDM) is summarized below:

i. **Candidate Sets Generation:** Generate candidate sets CGI(k) based on GLI(k-1), itemsets that are supported by the Si at the (k-1)th iteration, using the classic a priori candidate generation algorithm. Each site generates candidates based on the intersection of globally large (k- 1) itemsets and locally large (k- 1) itemsets.

ii. **Local Pruning:** For each X CGI(k), scan the database DBi at Si to compute X:supi. If X is locally large Si, it is included in the LLi set. It is clear that if X is supported globally, it will be supported in one site.

iii. **Support Count Exchange:** LLi(k)are broadcast and each site computes the local support for the items in ULLi(k).

iv. **Broadcast Mining Results***:* Each site broadcasts the local support for itemsets in ULLi(k). From this, each site is able to compute L(k).

3) Secure Multiparty Computation:

Substantial work has been done on secure multiparty computation. The key result is that a wide class of computations can be computed securely under reasonable assumptions. We give a brief overview of this work, concentrating on material that is used later in the paper. The definitions given here are from Goldreich. For simplicity, we concentrate on the two-party case. Extending the definitions to the multiparty case is straightforward.

1) Security in Semihonest Model:

A semihonest party follows the rules of the protocol using its correct input, but is free to later use what it sees during execution of the protocol to compromise security. This is somewhat realistic in the real world because parties who want to mine data for their mutual benefit will follow the protocol to get correct results. Also, a protocol that is buried formal definition of private two-party computation in the semihonest model is defined. Computing a function privately is equivalent to computing it securely.

2) Yao's General Two-Party Secure Function Evaluation

Yao‘s general secure two-party evaluation is based on expressing the function f(x,y) as a circuit and encrypting the gates for secure evaluation [3]. With this protocol, any twoparty function can be evaluated securely in the semihonest model. To be efficiently evaluated, however, the functions must have a small circuit representation. We will not give details of this generic method; however, we do use this generic result for securely finding whether a >=b (Yao‘s millionaire problem). For comparing any two integers securely, Yao‘s generic method is one of the most efficient methods known, although other asymptotically equivalent but practically more efficient algorithms could be used as well.

3) Commutative Encryption

Commutative encryption is an important tool that can be used in many privacy-preserving protocols. An encryption algorithm is commutative if the following two equations hold for any given feasible encryption keys k1, k2,... ., Kn K, any message M, and any permutations of i,j:
Eki1(…Ekin(M)…..)= Eji1(…Ejin(M)…..) M1,M2 M such that M1 M2 and for given k, <1 2k
Pr(Eki1(…Ekin(M)…..)= Eji1(…Ejin(M)…..))<
These properties of commutative encryption can be used to check whether two items are equal without revealing them.
For example, assume that party A has item iA and party B has item iB. To check if the items are equal, each party encrypts its item and sends it to the other party: In addition to meeting the above requirements, we require that the encryption be secure. Specifically, the encrypted values of a set of items should reveal no information about the items themselves.

SECURE ASSOCIATION RULE MINING

We will now use the tools described above to construct a distributed association rule mining algorithm that preserves the privacy of individual site results. The algorithm given is for three or more parties.

Method

Our method follows the general approach of the FDM algorithm, with special protocols replacing the broadcasts of LLi(k) and the support count of items LLi(k). We first give a method for finding the union of locally supported itemsets without revealing the originator of the particular itemset. We then provide a method for securely testing if the support count exceeds the threshold.

Secure Union of Locally Large Itemsets

In the FDM algorithm (Section 2.1.1), Step 3 reveals the large item sets supported by each site. To accomplish this without revealing what each site supports, we instead exchange locally large itemsets in a way that obscures the source of each itemset. We assume a secure commutative encryption algorithm with negligible collision probability
The main idea is that each site encrypts the locally supported itemsets, along with enough fake ‖ itemsets to hide the actual number supported. Each site then encrypts the itemsets from other sites. In Phases 2 and 3, the sets of encrypted itemsets are merged. Since (3) holds, duplicates in the locally supported itemsets will be duplicates in the encrypted itemsets and can be deleted. The reason this occurs in two phases is that if a site knows which fully encrypted itemsets come from which sites, it can compute the size of the intersection between any set of sites. While generally innocuous, if it has this information for itself, it can guess at the itemsets supported by other sites. Permuting the order after encryption in Phase 1 prevents knowing exactly which itemsets match; however, separately merging itemsets from odd and even sites in Phase 2 prevents any site from knowing the fully encrypted values of its own itemsets. Phase 4 decrypts the merged frequent itemsets. Commutativity of encryption allows us to decrypt all itemsets in the same order regardless of the order they were encrypted in, preventing sites from tracking the source of each itemset.

Association Rule:

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

Apriori Algorithm:

Apriori is designed to operate on databases containing transactions. The purpose of the Apriori Algorithm is to find associations between different sets of data. It is sometimes referred to as "Market Basket Analysis". Each set of data has a number of items and is called a transaction. The output of Apriori is sets of rules that tell us how often items are contained in sets of data.

Algorithm - Fast Distributed Mining (FDM)

The FDM algorithm proceeds as follows:

(1) Initialization
(2) Candidate Sets Generation
(3) Local Pruning
(4) Unifying the candidate item sets
(5) Computing local supports
(6) Broadcast Mining Results

III. CONCLUSION

We proposed a new efficient method in order to keep confidentiality of data in database. Our algorithm uses three methods of randomizing data (use of random values alongside *support values* of each *L.L-itemset*), anonymous sending of *M.L.L-itemset* and safe computation of *support values* of each *L.L-itemset*.

A virtue of this protocol compared with other protocols is that under an appropriate precision, security, and efficiency of our protocol is considerable without expensive coding mechanism. Through conspiracy of a maximum *n-2* sites, confidentiality of data of other sites is also protected. Furthermore, high flexibility is another advantage of our method, since each site based on its own trust on other sites can regulate the level of confidentiality of its data.

IV. FUTURE WORK

In proposed work delivery predictability is calculated by using three metrics as-number of encounters between nodes, time span between their meetings and transitive property of delivery predictability. It will be interesting to evaluate delivery predictability by using different metrics like context information and history of nodes.

V. REFERENCES

[1] Ch. Aggarwal; Ph. Yu, “Privacy-Preserving Data Mining: Models and Algorithms”, Kluwer Academic publishers ,2007.

[2] R. Agrawal; R. Srikant, “Fast algorithms for mining association rules,” Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, September, pp.487-499, 1994.

[3] M. Kantarcioglu; C. Clifton, “Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data”, IEEE Trans. Knowl. Data Eng. 16(9): 1026-1037, 2004.

[4] D. W.-L. Cheung; J. Han, V. Ng; A. W.-C. Fu; Y. Fu, “A fast distributed algorithm for mining association rules" Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS'96), Miami Beach, Florida, USA, Dec. 1996.

[5] X. Yi; Y. Zhang, Privacy-preserving distributed association rule mining via semi-trusted mixer, Data and Knowledge Engineering, page 550–567, 2007.

[6] Y. Lindell; B. Pinkas, “Privacy preserving data mining”, Advances in Cryptology, CRYPTO 2000 ,2000.

[7] Y. Lindell; B. Pinkas, “Secure Multiparty Computation for Privacy-Preserving Data Mining”, Journal of Privacy and Confidentiality, 2008.

[8] Ch.Ch Chang; J. Yeh; Y-Ch. Li, “Privacy-Preserving Mining of Association Rules on Distributed Databases”, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.11, 2006.

[9] A.A., Veloso; Jr.W. Meira; S. Parthasarathy; M.B. de Carvalho, “Efficient, accurate and privacy preserving data mining for frequent itemsets in distributed databases,” Proceedings of the Brazilian Symposium on Databases, Manaus, Amazonas, Brazil, pp.281-292, 2003.

[10] W. Du; M. J. Atallah, “Secure multi-party computation problems and their applications: A review and open problems”, In Proceedings of the 2001 New Security Paradigms Workshop, Cloudcroft, New Mexico, 2001.

[11] D. Chaum., "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms”. Communications of the ACM, 1981.

[12] M. Reiter; A. Rubin., “Crowd: Anonymity for Web Transaction”, ACM Transactions on Information and System Security, 1998.

[13] A. HajYasien, “Revisiting Protocol for Privacy Preserving Sharing Distributed Data: A Review with Recent Results”, Springer, pp. 542-555, 2011.