International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 5 126 – 129

Implementation paper on Novel Protocol for secure mining in Horizontally Distributed Database

Nidhi Kumari
Department of computer science Engineering
G. H. Raisoni Academic of Engineering and Technology,
Nagpur, India
singhnidhik@gmail.com

Prof. Sonali Bodkhe
Department of computer Science Engineering
G. H. Raisoni Academic Of Engineering and Technology
Nagpur, India
Sonali.mahure@gmail.com

***Abstract*—**Data mining is searching and extracting for knowledge in large repository, these collections sometimes are divided among various parties of the same database. Security concerns of knowledge may prevent the parties from directly sharing some types of information. This paper address a novel protocol for secure mining of association rule in horizontally distributed data base. This protocol uses fast data mining algorithm, which provide security while mining and producing relevant information. Main two parts of this protocol is secure multiparty algorithm- first one computes the unification of the each participating player's private subsets hold that hold by them, and other one tests an element's inclusion held by one player in a subset held by other player. This protocol is efficient for communicational cost and computational cost.

Keywords*—*frequent item-sets, association rule, secure mining, distributed data mining

*****

I. INTRODUCTION

Data mining technology is used to identify patterns and trends from large repository of data. Most tools operate in data mining by gathering or collecting all data into a central site and then running a mining algorithm against that gathered data. However, privacy concerns can prevent building a centralized warehouse- data may be distributed in several custodians, none of which is allowed to other site.

There are several players in horizontally partitioned database that hold homogeneous database. To find all association rules with support s and confidence c that hold in this unified database, also minimizing the information disclosed about the secured databases held by all party.

This paper we use a novel protocol to addresses the problem of secure multi-party computation of association rule in horizontally distributed database. In that the homogeneous database (i. e., the database hold information on different entities but all shares same schema) can be placed at several places, held by several parties. The goal is to minimizing disclosure of and providing security to the information gathered and accessed by from the private database. In this context the extracted information that we would like to keep secure is not only individual transaction in the partitioned databases, but also more global knowledge such as what association rule are supported locally in each of those database. The problem here, we provide the inputs which are partial database, and required outcome is the list of association rule that hold in the partitioned database which support s and confidence c which having no smaller then the given threshold value of each. The generic solution is a Boolean circuit which describe function f, for small inputs and it is the simple circuit which resize the function. For carrying out computation like this complex, such as ours, other methods are required.

Our novel protocol is based on two novel secure multiparty algorithms which provide enhanced security, privacy, efficiency as it uses commutative encryption. This protocol is based on: Fast data mining algorithm which is an unsecured version of the Apriori algorithm, computes two secure multiparty algorithms: 1. Computes the union of private subsets that each interacting players hold, 2. Tests the inclusion of an element held by one party in subset held by another.

In horizontally partitioned database there are several players that hold homogeneous database. Our Novel protocol offers enhanced privacy with respect to the current leading Kantarcioglu and Cliftons protocol,[1] more efficient in terms of computational cost, communication cost.

II. RELATED WORK

In previous work of mining has considered two related settings for private and secured data mining. In first one data miner and data owner are two different entities, and in second one, several parties hold their data who aim to jointly perform secure mining on the homogeneous (unified) corpus of data held by them.

To keep secure the data records from data miner is goal in the first setting. Hence prior to release of relevant information, the aim of data owner should be data anonymization by W. Jiang and C. Clifton [2]. In this context the main approach is to apply data perturbation. To infer general trends of data, the perturbed data can be used which prevent revealing of original record information.

In the second setting, by protecting data records of multiparty (i.e. each data owners) the goal is to perform data mining task. This is a problem of secure computation of multi-party. Here the usual cryptographic approach is as usual then probabilistic.

Previously there were various data mining techniques which ware proposed by various authors for security. M. Kantarcioglu and C. Clifton has proposed techniques for Privacy-preserving distributed mining of association rules on horizontally partitioned data [1]. While mining in the distributed databases privacy concerns may prevent the party from directly sharing the secured/private data. Jaideep Vaidya and Chris Clifton address the problem of mining of association rule, where transactions are distributed across sources for two-

party algorithm [3]. In [4] R Agrawal and R. Srikant proposed fast algorithm for mining association rules in large databases.

III. METHODOLOGY

a. Process Design

Let D be transaction database. We can view D as a binary matrix of R rows and C columns, where each row is a transaction over some set of items in suppose A= $\{a_1,a_2,a_3,\ldots,a_c\}$ and each columns in that matrix represents one of the items in A. This database is horizontally partitioned between some players P_1, P_2, P_3,…, P_M. The database D, is horizontally partitioned between M players. Each player holds partitioned (i.e., partial) database Dm that contains $N_m = |Dm|$ of the transactions in D, $1 \le m \le M$. the unified database is $D = D_1 U \ldots U D_m$ and it includes N_m where $1 \le m \le M$ transactions. Let subset of A has an item-set X. Whose global support, *supp(X)*, is the number of transactions in D that contains it. Its local support $supp_m(X)$, is the transaction number of D_m (Partial Database) that contained it. Clearly, *supp(X)* is all $supp_m(x)$ for all $1 \le m \le M$. Let s be a required support threshold. It is a real number between 0 and 1. An item-set X, is called s-frequent if $supp(X) \ge sN$. And for locally s-frequent of a partial database D_m if $supp_m(X) \ge sN_m$.

Our protocol here used is based on Fast Distributed Mining algorithm. This protocol is based on apriori algorithm which is unsecured version, means it shows partially data of patient but some of personal information is hidden to other party. Its main idea is for any site there must be one locally s-frequent itemset in order to find all globally item-sets. So each player has to reveal his locally s-frequent item-sets and to check whether it is globally s-frequent.

The fast data mining algorithm proceeds as follows:

Initialization:- Assume that the players have already jointly calculated s-frequent item-sets. To proceed and calculate global s-frequent.

(1) Candidate Set Generation:- Each player Pm computes the set of all (k-1)- item-sets that are locally frequent in his/her site and globally frequent; namely, Pm computes the local set which intersection with other players local set. He then applies on that set the Aproiri algorithm in order to generate set B_s of candidate k-item-sets.

(2) Local pruning: For each $X \in B_s$ of candidate k-item sets. Pm computes $supp_m(X)$. he then retains only those item-sets that are locally s-frequent. We denote this collection of item-sets by pruned item sets C_s for all k.

Unifying the candidate item-sets: Each player broadcasts global sets which is union of all local pruned itemsets.

(3) **Computing local Supports:** All players compute the local supports of all item-sets in one set.

(4) **Broadcasting results:** All player here broadcasts the local supports of item-sets that he computed. The FDM algorithm starts by finding all single items that are globally s-frequent item-sets. If the length of such item-sets is K, then in the $(K+1)^{th}$ iteration of the FDM it will find no (k+1)-item-sets that are globally s-frequent, where it terminates.

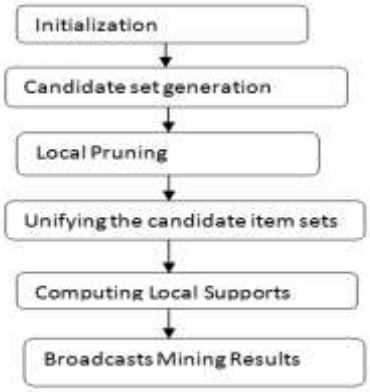


Fig. 1. Fast Data Mining Algorithm for Secure Mining

Input :- each player Pm $1 \le m \le M$ has an input set $FC_s^{k,m}$ is subset of the whole data.

Output:- C_s^k = unification of $C_s^{k,m}$.

Stage I: Start

1. All Player P_m $1 \le m \le M$, with their input decide on a commutative cipher and selects a random secrete encryption key km.
2. All players selects a hash function *h* and computes *h(x)* for all $x \in A_p (K_s^{k-1})$
3. Build a lookup table $T = \{ (x,h(x)) : x \in A_p (K_s^{k-1})$

Stage II: All item-sets encrypted here

4. For all player P_m $1 \le m \le M$, do
5. Set Xm = 0;
6. For all $x \in C_s^{k,m}$ do
7. All players computes $E_{km}(h(x))$ and adds it to X_m
8. End for
9. For I = 2 to M do
10. For all $1 \le m \le M$ do
11. P_m who sends permutation of X_m to P_{m+1}
12. P_m receives P_{m+1} the permutated X_{m-1}
13. P_m computes a new X_m as the encryption of the permuted X_{m-1} using the key K_m
14. End for
15. End for

Stage III

16. Each even/odd player sends his encrypted set to player P_2 /P_1 respectively
17. Player P_1 performs unification of all sets that were sent by odd players then check and remove duplicates.
18. Player P_2 performs unification of all sets that were sent by the even players then check and remove duplicates.
19. Player P_2 sends his permuted list of item-sets to P_1
20. P_1 unifies his list of item-sets and the received list from P_2 then duplications are removed from the unified list. Which is Denoted by final list by EC_s^k

Stage IV: Decryption

21. For m = 1 to M-1 do
22. P_m decrypts all item-sets in EC_s^k using k_m

 ISSN: 2321-8169
126 – 129

23. Pm sends the permuted (and Km-decrypted) EC_s^k to P_{m+1}
24. End for
25. Pm used the lookup table T to replace hashed values with the actual item-sets, and to identify and remove faked item-sets.
26. P_M broadcasts the result C_s^k

In the very first stage iteration of Fast Data Mining algorithm, when k=1, $C_s^{1,m}$ is the set that the m^{th} player would compute (stage 2-3) is just $F_s^{1,m}$, namely, the single item's set that are local s-frequent in database D_m. the complete Fast data mining algorithm starts by finding all single items that are globally s-frequent. It then proceeds to find for two item-sets that are globally s-frequent, and so forth until it finds the largest globally s-frequent item-sets. Here the length of such type of item-sets is k, then in the $(k+1)^{th}$ of the FDM Mining will find no $(k+1)^{th}$ –item-sets that are globally s-frequent, in that case it terminates Fast Data Mining Algorithm.

IV. Implementation Details

Here we took an example of hospital management, where two players hold partial database. First party not able to see other party's database and vice versa. Here role played of party is doctors. Doctors can enter their patient's data and one doctor cannot see other doctor's data. But all data are stored in the same database. It is horizontally distributed to all doctors. When the third party mines information from hospital database they can see only get public data, but some private data is hidden or not mined by third party. Here role of third party is government. They can see only those information which are public, and also while mining these data is in encrypted form. The very first stage iteration of Fast Data Mining algorithm, when k=1, $C_s^{1,m}$ is the set that the m^{th} player would compute (stage 2-3).

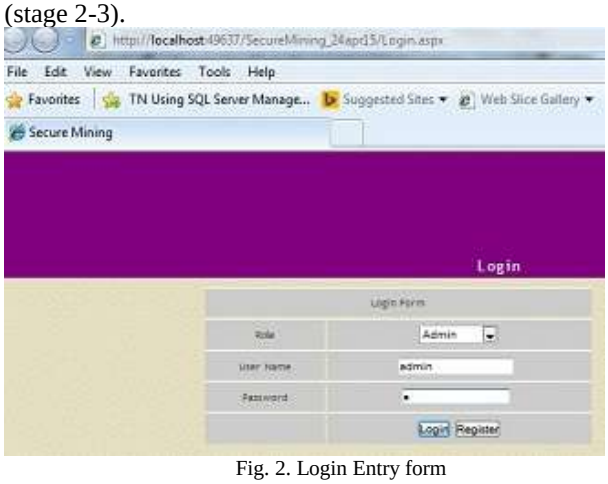


Fig. 2. Login Entry form

In the second figure above there is a login form, this form can be used by three role:- one is administration who manages homogeneous database. Second is doctors who plays role of multiparty, partial database can be managed by him, and third, government, who is data miner, some data excluding some secure information, can be mined by him.

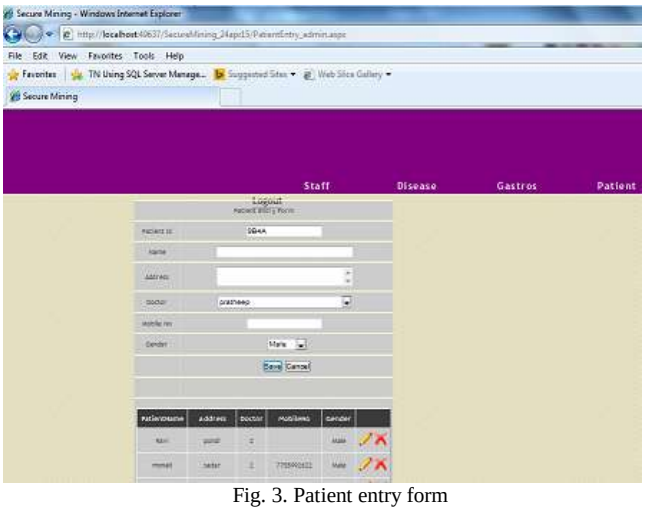


Fig. 3. Patient entry form

In the third figure entry of patient can be managed by players.

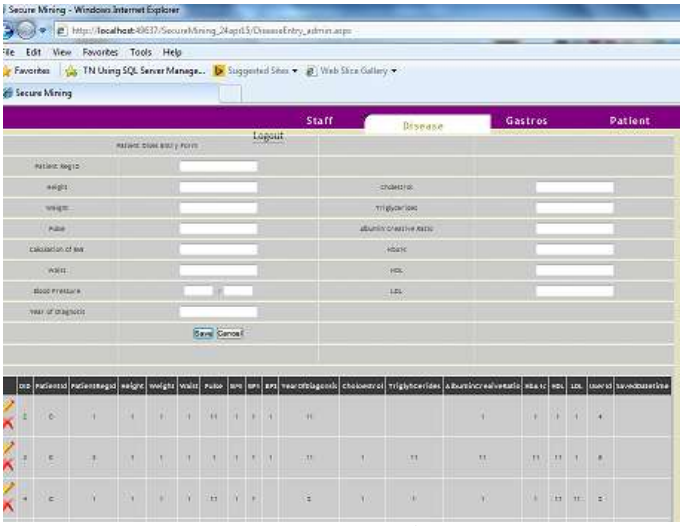


Fig. 4. Disease entry form

Forth figure above is very securely maintained by doctors to fill his part of database.

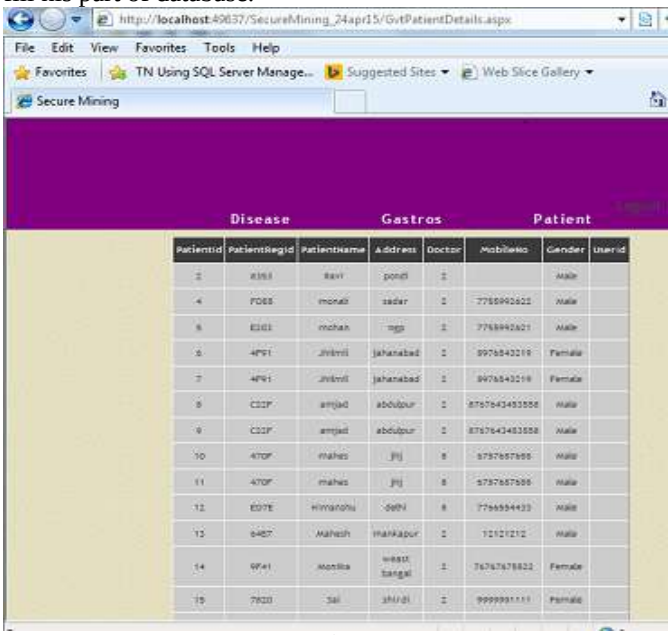


Fig. 5. Results of Third party miner

The above database shown is result of mined information, where some of disease fields are securely preserved by miner.

All experiments were implemented in C# (.net 4) and were executed on an Intel(R) Core(TM)i7-2620M personal computer with a 2.7 GHz CPU, 8 GB of RAM, and the 64-bit operating system Windows 7 Professional SP1.

V. RESULTS

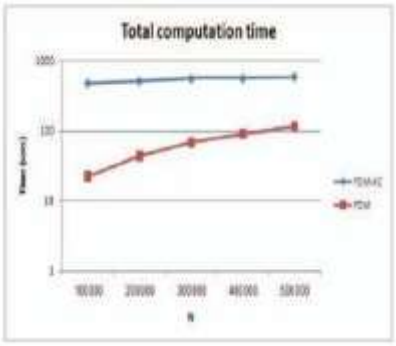
Consider database as D of N = 18 item-sets over a set of I = 5 items, as B= {1, 2, 3, 4, 5}. This D is partitioned between three players P = 3 and following is corresponding partial D1, D2, D3 databases:

$D1$ = {12, 12345, 124, 1245, 14, 145, 235, 24, 24}
$D2$ = {1234, 134, 23, 234, 2345}
$D3$ = {1234, 124, 134, 23}.

For example, if D1 includes transactions N1=9, its third transaction in lexicographic order here consists 3 items i.e. 1, 2, 4. So threshold support (s) = 0.33 (i.e. 1/3) for $0 < s \leq 1$ and confidence support c = 0.33 for $0 < c \leq 1$.

Table 1 Result set using FDM and Unifi KC Algorithm:

SR. NO	Item-Set	Support Value	Confidence Value
1	1	0.61	-----
2	2	0.78	-----
3	3	0.56	-----
4	4	0.78	-----
5	1,2 = {1} -> {2}	0.39	0.64
6	1,4 = {1} -> {4}	0.56	0.92
7	2,3 = {2} -> {3}	0.44	0.56
8	2,4 = {2} -> {4}	0.56	0.72
9	3,4 = {3} -> {4}	0.39	0.7
10	1,2,4 = {1} -> {2,4}	0.33	0.54
11	1,2,4 = {1,2} -> {4}	0.33	0.85



V. RELATED WORK

Previous work in securely preserved data mining considered two related settings in first data miner and data owner are two different entities and other several party holds distributed data and they aim jointly perform data mining on the unified collection of data they hold.

In the first setting protects information from miner, hence, player's aims at annonimizing data before its release. For this approach data perturbation was applied.

And in second setting, the goal is to mine data as well protecting data records of each of the owners of data. This is called secure multi-party computation where cryptographic approach is used usually as compared to probabilistic. Lindell and Pinkas showed how to securely perform an ID3 decision tree when the training set is horizontally distributed.

VI. CONCLOSION

We proposed a novel secure protocol for secure mining of association rule in database which is horizontally distributed that significantly improves upon leading protocol in terms of efficiency, security and privacy. In our novel secure protocol one of the most ingredient multiparty protocol for computing the unification (i. e. union) and intersection, of private subset that is held by each participating players. And other ingredients are a protocol that tests the inclusion of item-sets held by one player in a subset held by other players.

VI. ACKNOWLEDGMENT

We wish to thank my guide Prof. Sonali Bodkhe for her insightful comments and suggestions.

VII. REFERENCES

[1] Tamir tassa, "Secure Mining of Association Rules in Horizontally Distributed Databases", IEEE transactions on knowledge and data engineering, 2013.

[2] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 23-26 2002, pp. 639–644.

[3] M.Kantarcioglu and C. Clifton., "Privacy-preserving distributed mining of association rules on horizontally partitioned data", *IEEE Transactions on Knowledge and Data Engineering*, 16:1026–1037, 2004.

[4] R.Agrawal and R. Srikant.,"Privacy-preserving data mining", *SIGMOD Conference*, pages 439–450, 2000.

[5] A.V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules", In *KDD*, pages 217–228, 2002.

[6] M. Kantarcioglu, R. Nix, and J. Vaidya,"An efficient approximate protocol for privacy-preserving association rule mining", In *PAKDD*, pages 515– 524, 2009.

[7] M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold., "Keyword search and oblivious pseudorandom functions", In *TCC*, pages 303–324, 2005.

[8] R. Srikant and R. Agrawal. Mining generalized association rules. In VLDB, pages.