SECURE MINING OF ASSOCIATION RULES IN HORIZONTALLY AND VERTICALLY DISTRIBUTED DATABASES

F. A. Patel[1], Prof. A. S. Tamboli[2], Prof. S. P. Patil[3]
[1]Student, Computer Science and Engineering, Annasaheb Dange College of Engineering & Technology Ashta, India
[2]Asst. Professor, Computer Science and Engineering, Annasaheb Dange College of Engineering & Technology Ashta, India
[3] Asst. Professor, Computer Science and Engineering, Annasaheb Dange College of Engineering & Technology lAshta, India

Abstract : ***In this paper we propose a protocol for secure mining of association rules in horizontally and vertically distributed databases. The current leading protocol is that of Kantarcioglu and Clifton[6] .This protocol like theirs, it is based on the Fast Distributed Mining (FDM) algorithm of Cheung etal[7]., which is an unsecured distributed version of the Apriori algorithm. The main goal of this protocol are two novel secure multi-party algorithms—one that computes the union of private subsets that each of the interacting players hold, and another that tests the inclusion of an element held by one player in a subset held by another. Our protocol provide enhanced privacy with respect to the other protocol . In addition, this horizontally and vertically distributed database is simpler and is significantly more efficient in terms of communication rounds and computational cost.***

Keywords- Privacy preserving data mining, frequent item sets, association rules, Apriori algorithm

1. INTRODUCTION

Data mining technology has emerged as means of identifying patterns and trends from large data. Data mining is applied on data from large database. Now a days database is distributed among organization. Organization needs global knowledge of data for which they have to share information among different users but privacy concern may prevent them from sharing information. For this purpose secure computation is required.

The goal of secure computation is to enable different parties each with its own private input, to compute a function of their joint inputs without revealing any information except for the value of the function. In secure multi-party computation there are M players that hold private inputs, x_1 ... $.x_M$, and they wish to securely compute y= f(x_1. . . . $.x_M$) for some public function f. If there existed a trusted third party, the players could surrender to him their inputs and he would perform the function evaluation and send to them the resulting output.

In the absence of such a trusted third party, it is needed to devise a protocol that the players can run on their own in order to arrive at the required output y. Such a protocol is considered perfectly secure if no player can learn from his view of the protocol more than what he would have learnt in the idealized setting where the computation is carried out by a trusted third party.

In this paper we proposed a protocol for secure mining of association rules in horizontally and vertically distributed databases. The goal is to find all association rules with support at least s and confidence at least c, for some given minimal support size sand confidence level c, that hold in the unified database, while minimizing the information disclosed about the private databases held by those players.

2. RELATED WORK

The information that we would like to protect in this context is not only individual transactions in the different databases, but also more global information such as what association rules are supported locally in each of those databases. That goal defines a problem of secure multi-party computation. In such problems, there are M players that hold private inputs, x_1; . . . ; x_m, and they wish to securely compute y=*f* (x_1; . . . ; x_m) for some public function f. If there existed a trusted third party, the players could surrender to him their inputs and he would perform the function evaluation and send to them the resulting output. In the absence of such a trusted third party, it is needed to devise a protocol that the players can run on their own in order to arrive at the required output y. Such a protocol is considered perfectly secure if no player can learn from his view of the protocol more than what he would have learnt in the idealized setting where the computation is carried out by a trusted third party. Yao[2] was the first to propose a generic solution for this problem in the case of two players. Other generic solutions, for the multi-party case, were later proposed in [3], [4], [5].

In our problem, the inputs are the partial databases, and the required output is the list of association rules that hold in the unified database with support and confidence no smaller than the given thresholds s and c, respectively. Kantarcioglu and Clifton studied that problem in [6] and devised a protocol for its solution. In this paper they addresses the problem of computing association rules in a scenario where data may be distributed among several custodians, none of which are allowed to transfer their data to another site. All sites have the same schema, but each site has information on different entities. The goal is to produce association rules that hold globally, while limiting the information shared about each site. In this paper method has proposed ,which follows the general approach of Fast distributed mining of association rules algorithm proposed in[4] with special protocol replacing broadcasting of locally frequent item set and support count of item in locally frequent item set.

In this paper two protocol has proposed Protocol 1, privately computes the union of the locally large item set and protocol 2 is proposed to find global support count securely.In protocol 1 there are total 5 phases. In Phase 0 encryption of locally frequent item set at each site is performed and size of locally frequent item set is equal to the size of global frequent item set of previous round. In Phase 1

encryption by all site is performed. There are totally N-1 rounds during first round each site I sends permuted locally frequent item set to site (i+1)mod N. In next rounds each site I encrypts the received item sets from other sites by using encryption key and send that to site (i+1)modN. In phase 2 site 0 receives encrypted item sets from all even sites. Site 0 sets Ruleset_1 : $U_{j=1}^{\lceil (N-1)/2 \rceil} \text{LL}_{e(2j-1)(k)}$ and site1 sets Ruleset_0: $U_{j=0}^{\lfloor (N-1)/2 \rfloor} \text{LL}_{e(2j)(k)}$. In phase3 site 1 sends permuted Ruleset_1 to site 0 and site0 sets Ruleset which is union of Ruleset_0 and Ruleset_1. In phase4 for N-1 rounds each site decrypts the items in Ruleset and sends permuted Ruleset to site (i+1)mod N and finally N-1 site decrypts items in Ruleset and broadcast Ruleset to other sites. Protocol2 find global support count securely.

This protocol and its implementation relies upon cryptographic primitives such as commutative encryption, oblivious transfer, and hash functions. Protocol proposed in this paper is improved version of protocol proposed in [6].

3. PRELIMINARIES

3.1 Some of the Definitions and Notations

- Let D be a transaction database. In D each row is a transaction over some set of items, A = f{a_1.....a_L}where each column represent one of the item in A. Database D is partitioned between M players denoted by P_1, ..., P_M. Player P_M holds the partial database D_m that contains N_M= $|D_M|$ of transaction in D, 1≤m≤M. The unified database D is, D=D_1 U.....U D_M.

- An item set X is a subset of A. Its global support, supp(X), is the number of transactions in D that contain it. Its local support, supp_m(X), is the number of transactions in D_m that contain it. Clearly supp(X)= $\sum_{m=1}^{M} S\text{upp}_m(X)$.
- Let s be a required support threshold. An item set X is called s-frequent if supp(x)≥ sN where N is no. of transaction in database. It is called locally s-frequent at D_m if $\text{supp}_m(x) \geq sNm$
- For each 1 ≤ k ≤ L(L is no. of columns in tansaction database),let F_s^k denote the set of all k-item sets (namely, item sets of size k) that are s-frequent, and $F_s^{k,m}$ be the set of all k-item sets that are locally s-frequent at Dm,1≤m≤M.

4. SECURE MINING OF ASSOCIATION RULE

The proposed protocol is depend on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [7]

4.1 Secure Computation Of All Locally Frequent Item Sets

In Secure Computation of Locally Frequent Item Sets each player securely computes the locally frequent itemset. Proposed protocol is based on the Fast Distributed Mining (FDM) [7] algorithm which is an unsecured distributed version of the Apriori algorithm.Its main idea is that any s-frequent item set must be also locally s-frequent in at least one of the sites. Hence, in order to find all globally s-frequent item sets, each player reveals his locally s-frequent item sets and then the players check each of them to see if they are s-frequent also globally.

Protocol UNIFI-Unified list of all locally frequent item sets

Algorithm-

Input : Each player P_m has input set $C_s^{k,m} \subseteq \text{Ap}(F_s^{k-1})$, $1 \leq m \leq M$

Output : $C_s^k = \cup_{m=1}^{M} C_s^{k,m}$

Steps:

[1]. Each players select the needed cryptographic primitives and each player selects a corresponding private random key and select a hash function h to apply on all item sets prior

[2]. Each player P_m hashes all item sets in C_s^k and then encrypts them using the key K_m.Then, he adds to the resulting set faked item sets in order to hide the number of locally frequent item sets that he has. Resulting set is denoted by X_m

[3]. Each odd player sends his encrypted itemset to odd admin and each even player send his encrypted itemset to even admin.

[4]. Odd admin and even admin unifies all set that were sent by odd player and even player and after that they remove the duplicate from unified list.

[5]. Even admin sends his permuted list of itemset to odd player

[6]. Odd admin unifies his list of itemset and itemset received from even player and then removes duplicate from unified list.

[7]. Odd admin decrypts itemset from unified list and broadcast the result.

4.2 Identifying globally frequent itemset

.After finding locally frequent itemset we proced to find globally frequent itemset. In order to reveal which of the locally frequent item sets is globally s-frequent there is a need to securely compute the support of each of those item sets.

To find globally frequent itemset we use below formula. Let x be one of the candidate item sets in C_s^k. Then x is globally s-frequent if and only if

$\Delta(x) = \text{supp}(x) - sN = \sum_{m=1}^{M} (supp\text{m}(x) - sNm \geq 0$

Where s is required support and N is no of transaction.

4.3 IDENTIFYING ALL (S,C)-ASSOCIATION RULES

Once the set Fs of all s-frequent item sets is found, then we proceed to look for all (s, c) -association rules (rules with support at least sN and confidence at least c). For X,Y∈ Fs, Fs is set of items which are s frequent and X ∩Y=ϕ then corresponding association rule X⇒Y has confidence c if and only if supp(X ∪ Y) / supp(X) ≥ c. For deriving association rules in an efficient manner we will use the lemma X⇒ Y is an association rule and Y' is subset of Y then X ⇒ Y' is also an association rule.

5. Experimental Results

The database that we used in our experiment is synthetic database. We compared our performance with FDM and FDM-KC. The performance is measured depending on three parameter-

- N-the number of transactions in the unified database,
- M-the number of players, and
- s-the threshold support size.

In the first experiment set, we kept M and s fixed and tested several values of N. In the second experiment set, we kept N and s fixed and varied M. In the third set, we kept N and M fixed and varied s.

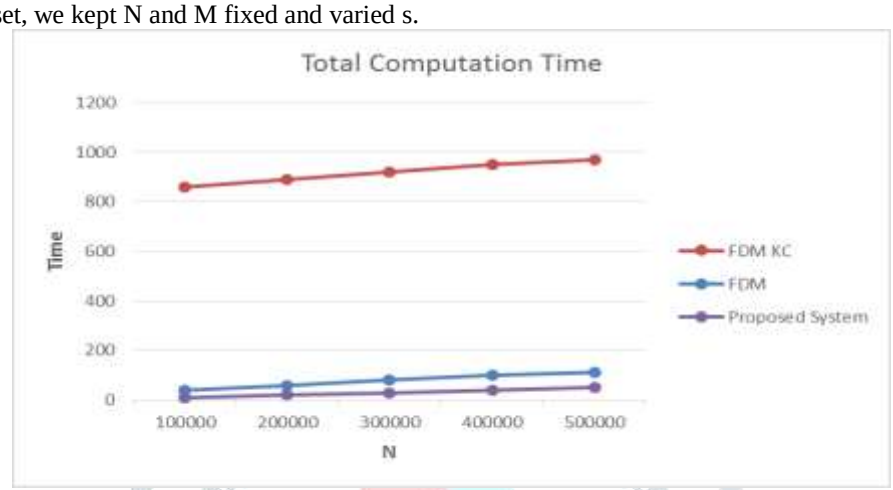

figure. 1

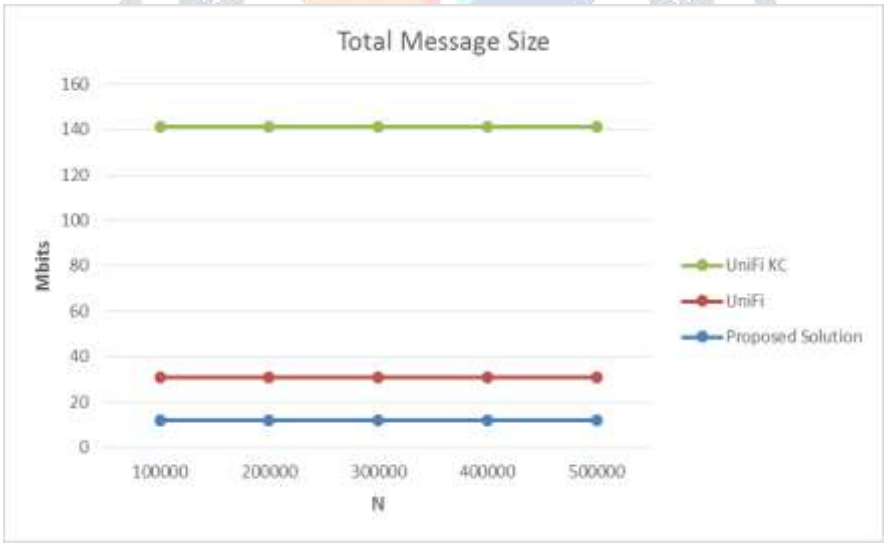

figure. 2

Above figure 1and 2 shows Computation and communication costs versus the number of transactions N for FDM-KC, FDM and proposed system. In this we kept M=10 and s=0.1.

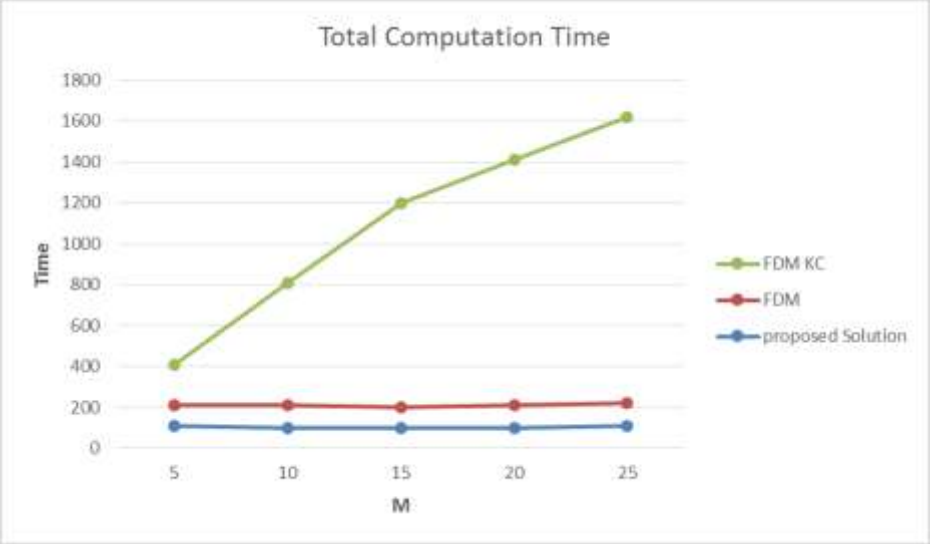

figure 4.

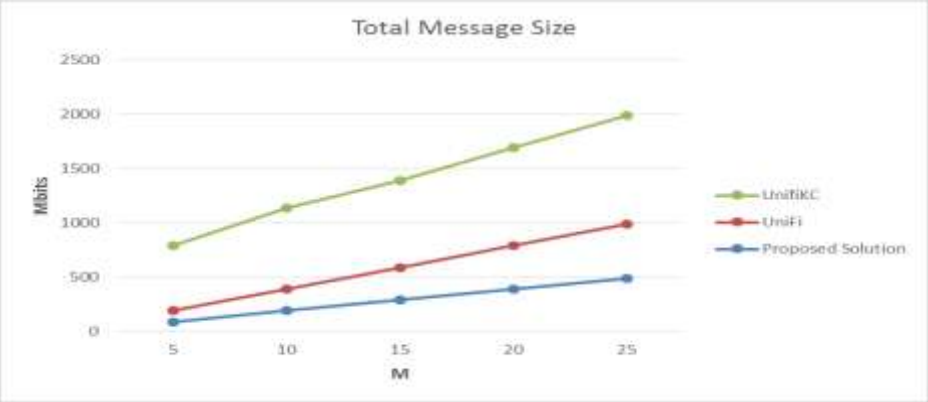

figure 5.

figure 4 and figure 5. shows communication and computation cost as function of M. In this experiment we kept N=5000000 and s=0.1. The experiments shows that the computation and communication costs increase with M.

REFERENCES

[1] Tamir Tassa, "Secure Mining of Association Rules in Horizontally Distributed Databases," IEEE transactions on knowledge and data engineering, vol. 26, no. 4, april 2014.

[2] A.C. Yao, "Protocols for Secure Computation," Proc. 23rd Ann. Symp. Foundations of Computer Science (FOCS), pp. 160-164, 1982.

[3] D. Beaver, S. Micali, and P. Rogaway, "The Round Complexity of Secure Protocols," Proc. 22nd Ann. ACM Symp. Theory of Computing (STOC), pp. 503-513, 1990.

[4] A. Ben-David, N. Nisan, and B. Pinkas, "FairplayMP - A System for Secure Multi-Party Computation," Proc. 15th ACM Conf. Computer and Comm. Security (CCS), pp. 257-266, 2008.

[5] O. Goldreich, S. Micali, and A. Wigderson, "How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority," Proc. 19th Ann. ACM Symp. Theory of Computing (STOC), pp. 218-229, 1987.

[6] M. Kantarcioglu and C. Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data,"IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1026-1037, Sept. 2004

[7] D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu, "A Fast Distributed Algorithm for Mining Association Rules," Proc. Fourth Int'l Conf. Parallel and Distributed Information Systems (PDIS), pp. 31-42, 1996.