Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 7, July 2014, pg.602 – 606

RESEARCH ARTICLE

A Novel Approach for Secure Mining of Horizontally Distributed Databases

Wasudev.W.Pingle, Prof. S.S. Banait
Dept. Of Computer Engg
K.K.W.I.E.E.R, Nashik
University of pune, Maharastra
Wpingle@gmail.com; ssbanait@kkwagh.edu.in

*Abstract— **The current protocols for secure mining of association rules in horizontally distributed databases has different Problems like privacy. In this kind of scenario Different players may wish to protect & preserve their data and metadata about their information. The approach is based on the Fast Distributed Mining (FDM) algorithm which is devised by Cheung [5]. FDM is distributed version of the Apriori algorithm and unsecured too. Most importantly these protocols provides two fresh secure algorithms (sub protocols) — the union of private subsets that each of the participating players hold is carried out by one and other one checks for the inclusion of an element held by one of the player in a subset which is held by another.***
Our systems provides optimized privacy with respect to the protocol in [7] using Symmetric Key Cryptography. In addition, relatively it is more efficient in terms of computational cost, communication rounds and also in communication cost because of MAP/REDUCE

Keywords— Association Rules, Fast Distributed Mining, Frequent Item-sets, Horizontally Distributed Databases, Map Reduce, Privacy

I. INTRODUCTION

Data mining technology has emerged as a means of identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand: most tools operate by gathering all data into a central site, then running an algorithm against that data. However, privacy concerns can prevent building a centralized warehouse – data may be distributed among several sites, none of which are allowed to transfer their data to another site. There are so many examples of applications of multi-party computation. Two of them are auctions and elections. Nowadays, trusted third party performs most of the computations (e.g. an auctioneer). To enable running of such applications without any third party use of secure multi-party protocols can be made. Tremendous work has been done on secure multi-party protocols. Yao has worked on secure two-party computation protocol and on secure multi-party computation; there have been many theoretical constructions of secure computation protocols. However, there are almost no system exists which enable programmers who are not experts in the theory of secure computation to implement such protocols. Consequently, if one desires to devise a system for secure computation then he/she may need to read all the relevant papers and implement the system from the scratch. This requirement imposes a huge barrier for the one who wishes to use secure computation. Our goal is to provide a secure way for multi-party computation and change above stated situation.

II. MINING ASSOCIATION RULES ON FREQUENT ITEMSETS

Progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as the basket data. A record in such data typically consists of the transaction date and the items bought in the transaction. Successful organizations view such databases as important pieces of the marketing infrastructure. They are interested in instituting information-driven marketing processes, managed by database technology, that enable marketers to develop and implement customized marketing programs and strategies .The problem of mining association rules over basket data was introduced. An example of such a rule might be that 98% of customers that purchase tires and auto accessories also get automotive services done. Finding all such rules is valuable for cross- marketing and attached mailing applications. Other applications include catalog design, add-on sales, store layout, and customer segmentation based on buying patterns. The databases involved in these applications are very large. It is imperative, therefore, to have fast algorithms for this task.

The following is a formal statement of the problem:

Let I = { i1 ; i2; . . ...; i_m }g be a set of literals, called items.

Let D be a set of transactions, where each transaction T is a set of items such that T is subset of I .

Associated with each transaction is a unique identifier, called its TID. We say that a transaction T contains X, a set of some items in I , if X is subset of I .

An association rule is an implication of the form X => Y, and X ∩ Y = Ø. The rule X => Y holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y .

The rule X => Y has support s in the transaction set D if s# of transactions in D contain X and Y .

Given a set of transactions D, the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support called minsup and minimum confidence called minconf respectively. Our discussion is neutral with respect to the representation of D. For example, D could be a data table, a relational table, or the result of a relational expression.

This is known as Apriori Algorithm.

III. FAST DISTRIBUTED MINING

As stated earlier in our protocol we are using Fast Distributed Mining (FDM) algorithm of Cheung et al. [5], which in simply unsecured distributed version of Apriori algorithm. Its main idea is that any itemset which frequent *s* times must be frequent locally at least *s* times. So each player reveals its local *s*-frequent itemset in order to know global *s*-frequent item-sets and after that all of them are checked to make sure if they are *s*-frequent globally.

Steps in FDM algorithm:-

Step I: Initialization: FDM algorithm start with the assumption that all the participating players have collectively calculated set of all *k*-itemset F_s^{k-1} (itemset of size k) which are *s*-frequent. Calculating F_s^k will be the next step.

Step II: Candidate Set Generation: Every player P_s calculates set of item-sets that are *s*-frequent locally as well as globally. Resulting itemset would be named as $F_s^{k-1,m} \cap F_s^{k-1}$. Later on Apriori algorithm is applied in order to form $B_s^{k,m}$ of item-sets from candidate.

Step III: Local Pruning: Player P_s computes $\square_m(X)$ for each $X \in B_s^{k.m}$ and only locally *s*-frequent item-sets are kept. This group of itemset can be represented by $C_s^{k,m}$

Step IV: Unification of Candidate item-sets: After broadcasting $C_s^{k,m}$ each player calculates $C_s^k \cup_{m-1}^{M} C_s^{k,m}$.

Step V: Computing Local Supports: All players compute supports of all item-sets of C_s^k that are local.

Step VI: Broadcast Mining Result: Now every participating player can derive global support for every itemset in C_s^k after broadcasting local supports. Finally we get F_s^k which is subset of C_s^k containing s-frequent item-sets globally.

IV. PRIVACY-PRESERVING DISTRIBUTED MINING OF ASSOCIATION RULES ON ITEMS

Kantarcioglu and Clifton [7] proposed a protocol which is protocol1 that computes unique list of all the local frequent item-sets $C_s^k = \cup_{(m=1)}^{M} C_s^{(k,m)}$. $C_s^k = \cup_{m=1}^{M} C_s^{k,m}$ This protocol does not expose content and size of the subsets $C_s^{k,m}$ $C_s^k = \cup_{m=1}^{M} C_s^{k,m}$ Protocol works on the assumption that every participating player knows upfront F_s^{k-1} which is list of all item-sets that are *s*-frequent globally and each player tend to compute F_s^k. Protocol of Kantarcioglu and Clifton can be referred as UNIFI-KC (Unifying list of all frequent item-sets (local) - Kantarcioglu and Clifton).

The main part of the protocol is a sub-protocol for the secure computation of the union of private subsets that are held by the different players. (The private subset of a given player, as we explain below, includes the item-sets that are s-frequent in his partial database.) That is the most costly part of the protocol and its implementation relies upon cryptographic primitives such as commutative encryption, oblivious transfer, and hash functions. This is also the only part in the protocol in which the players may extract from their view of the protocol information on other databases, beyond what is implied by the final output and their own input. While such leakage of information renders the protocol not perfectly secure, the perimeter of the excess information is explicitly bounded and it is argued there that such information leakage is innocuous, whence acceptable from a practical point of view.

Initially all player include some fake item-sets to their subsets so that no other player will know the actual size of their subset. Then, collectively all of them encrypt private subset by applying commutative encryption. Commutative encryption means adding encryption at each level by using private secret key. Commutative encryption ensures that all the item-sets in each of the subset are encrypted in same manner. Then players compute union of subsets which are encrypted. At last, decryption is carried out on union set and fake item-sets are removed.

V. PROPOSED SYSTEM

Our solution is based on Fast Distributed Mining algorithm using Centralized approach.

Master system will horizontally distribute the database among the slave systems by checking status (active/deactive) . The distribution will be securely managed using symmetric key cryptography algorithm.

The slave systems will collect the horizontal part of the database and generate frequent item-sets locally and subsequently mine the strong association rules. Slaves will then transfer back the results i.e association rules securely using symmetric key cryptography algorithm.

The master will be responsible for aggregation of those results and derive strong association rules which support global threshold. The entire process will include two programming paradigms- MAP and REDUCE.

Our protocol is purely independent of oblivious transfer and commutative encryption which makes it simple and moreover contributes to the relatively decreased cost of computation and communication

VI. DATA SET

We used synthetic database for our experimentation. We explained how the database was split horizontally into partial databases. The databases that we will be going to used for our experimental evaluation are synthetic databases that were generated using the same parameters that were introduced in [7] and then used also in Subsequent studies such as [6] [8] [14]. Following list gives the parameters that were used in generating the synthetic database.

1. Number of transactions in the whole database
2. Number of items
3. Transaction average size
4. Average size of maximal potentially large itemsets
5. Number of maximal potentially large itemsets
6. Clustering size
7. Pool size
8. Correlation level
9. Multiplying factor

For Generating this dataset we used tool IBM quest generator tool.

VII. ARCHITECTURE AND WORKING

Our prime focus will be on finding every association rule with minimum support s and minimum confidence c, for some finite support size s and some finite confidence level c, that reside in that particular database, without risking privacy of database held at various sites. The information that we are trying to secure is not just the local transactions in underlying databases of horizontally partitioned database as well as global information like what type of different association rules are in use locally in every individual database.

The main idea is to use the Map Reduce model is to hide details of parallel execution and allow users to focus only on data processing strategies along with symmetric key cryptography algorithm.

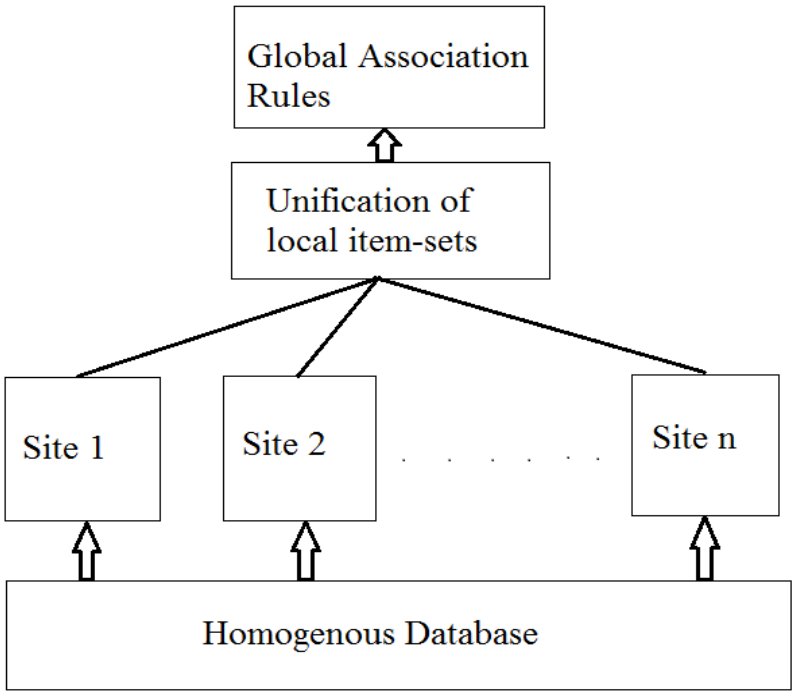

Fig 1. Architecture of the system

1. Data is horizontally partitioned over different sites securely using AES(Advanced Encryption Standard) i.e. databases that share the same schema but hold different information on objects.
2. Each site has complete information on a set of objects
3. Same attributes at each site but information is different.
4. Each site decrypts the data and find its locally frequent itemsets by using local date mining approach to generate these locally frequent itemsets by using Apriori algorithm.
5. Compute the union of the locally large candidate item sets securely using AES algorithm.
6. At the end check the confidence of the potential rules securely. The goal is to find association rules with support at least s and confidence at least c, for given minimal support size s and confidence level c, that hold in the database, while minimizing the information disclosed about the private databases held by those sites.

VIII. RESULT ANALYSIS

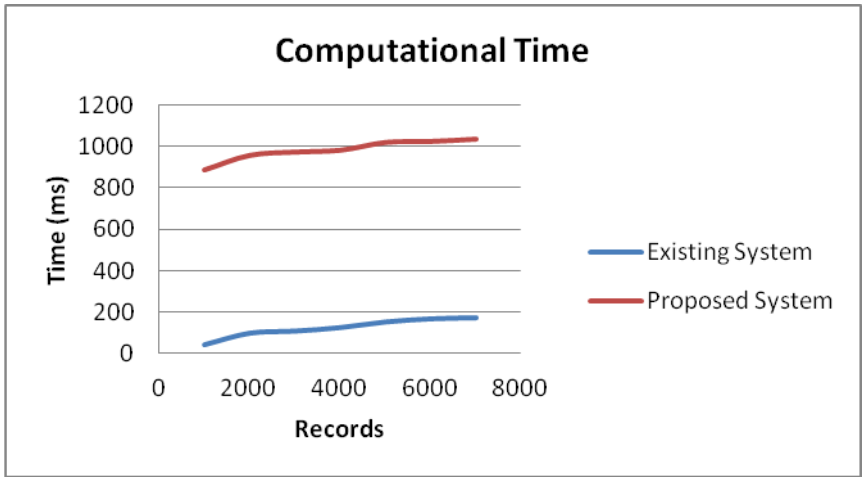

Fig1. Computation time to Calculate Frequent Itemsets

The Proposed system takes significantly lesser time as compared to the existing system. The proposed system is much more efficient compared to the existing system which is evident form the graph depicted.

IX. CONCLUSION

This system will provide alternatives which will make it possible to securely generate strong frequent itemsets to calculate association rules in distributed environment with maximum privacy with optimum performance. Our system is purely independent of oblivious transfer and commutative encryption which makes it simple and moreover contributes to the relatively decreased cost of computation and communication.

REFERENCES

[1] A.C. Yao. Protocols for secure computation. In *FOCS, pages* 160–164,982.
[2] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC*, Pages 218–229, 1987.
[3] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In *STOC*, pages 503–513, 1990.
[4] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for Message authentication. In *Crypto*, pages 1–15, 1996.
[5] 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. In *PDIS*, pages 31– 42, 1996.
[6] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *KDD*, pages 639–644, 2002.
[7] 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.
[8] Secure Mining of Association Rules in Horizontally Distributed Databases. Tamir Tassa 2013.
[9] S. Zhong, Z. Yang, and R.N. Wright. Privacy-enhancing k-anonymization of customer data. In *PODS*, pages 139–147, 2005.
[10] A. Ben-David, N. Nisan, and B. Pinkas. Fairplay MP - A system for secure multi-party computation. In *CCS*, pages 257–266, 2008.
[11] T. Tassa and D. Cohen. Anonymization of centralized and distributed social networks by sequential clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2012.
[12] T. Tassa and E. Gudes. Secure distributed computation of anonymized views of shared databases. *Transactions on Database Systems*, 37, Article 11, 2012.
[13] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB, pages 487–499 1994.
[14] Secure Mining of Association Rules in Horizontally Distributed Databases.Tamir Tassa
IEEE Transactions on Knowledge and Data Engineering VOL:PP NO:99 YEAR 2013

