International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 2 143– 146

Association Rules Mining in Distributed Databases

Shubhangi A. Ramteke
Computer Science and Engineering
Rajiv Gandhi College of Engineering, Research & Technology
Chandrapur, India
shubhangi_ramteke@ymail.com

Abstract— Here we proposed a protocol for mining of association rules in distributed databases in a secure manner. Different sites contains homogeneous databases i.e. databases shares the same schema but hold information on different entities. The goal is to find all association rules with support s and confidence c, while minimizing information disclosed about the private databases that is hold by those sites. There is need to find such a protocol for mining of association rules in horizontally distributed databases in a secure manner. Our protocol is based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [5], which is an unsecured distributed version of the Apriori algorithm. The main contents in our protocol re two secure multiparty algorithms- one that compute the union of private subset that each of the interacting sites holds , and another that tests the inclusion of an element held by one site in a subset held by another. Our protocol offers enhanced privacy, more efficient and simpler in terms of communication round, communication and computational costs with respect to the protocol in [16].

Keywords- *Fast Distributed Mining, Association Rules, UNIFI-KC (Unifying lists of locally Frequent Itemsets —Kantarcioglu and Clifton).*

*****

I. INTRODUCTION

The protocol that we propose here computes a parameterized family of functions, which we call threshold functions, in which the two extreme cases correspond to the problems of computing the union and intersection of private subsets.For this we provide input as partial databases and the required output is the list of association rules that hold in the unified database with support and confidence no smaller that our protocol may leak is less sensitive than the excess information leaked by other protocol. Different sites contains homogeneous databases i.e. databases shares the same schema but hold information on different entities. The goal is to find all association rules with support s and confidence c, while minimizing information disclosed about the private databases that is hold by those sites. There is need to find such a protocol for mining of association rules in horizontally distributed databases in a secure manner. Our protocol is based on the Fast Distributed Mining(FDM) algorithm which is an unsecured distributed version of the Apriori algorithm. The main contents in our protocol re two secure multiparty algorithms- one that compute the union of private subset that each of the interacting sites holds , and another that tests the inclusion of an element held by one site in a subset held by another. Our protocol offers enhanced privacy than previous protocol. In addition, it is more efficient and simpler in terms of communication round, communication and computational costs.

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, $x_1, \ldots, x_M$, and they wish to securely compute $y = f(x_1, \ldots, x_M)$ 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. Here we propose an alternative protocol for the secure computation of the union of private subsets. The proposed protocol improves upon simplicity and efficiency as well as privacy. In particular, our protocol does not depend on commutative encryption and oblivious transfer. While our solution is still not perfectly secure, it leaks excess information only to a small number of possible coalitions, unlike the previous protocol that discloses information also to some single sites. In addition, we claim that the excess information that our protocol may leak is less sensitive than the excess information leaked by the already existing protocol such as protocol of [16].

II. LITERATURE REVIEW

Previous work in privacy preserving data mining has considered two related settings. One, in which the data owner and the data miner are two different entities, and another, in which the data is distributed among several parties who aim to jointly perform data mining on the unified corpus of data that they hold. In the first setting, the goal is to protect the data records from the data miner. Hence, the data owner aims at anonymizing the data prior to its release. The main approach in this context is to apply data perturbation [2], [8].

The idea is that the perturbed data can be used to infer general trends in the data, without revealing original record information. In the second setting, the goal is to perform data mining while protecting the data records of each of the data owners from the other data owners. This is a

problem of secure multiparty computation. 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. We present here a protocol for computing that function which is much simpler to understand and program and much more efficient than those generic solutions. It is also much simpler than Protocol UNIFI-KC and employs less cryptographic primitives. Our protocol computes a wider range of functions, which we call threshold functions.

II. REASONS FOR SELECTING THE PROBLEM

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. Fast Distributed Mining (FDM) algorithm of Cheung et al is an unsecured distributed version of the Apriori algorithm. The FDM algorithm violates privacy in two stages-first is that where the site broadcast the itemsets that are locally frequent in their private databases, and other is , where they broadcast the sizes of the local supports of candidate itemsets. Kantarcioglu and Clifton proposed secure implementations of those two steps. Our improvement is with regard to the secure implementation, which is the more costly stage of the protocol, and the one in which the protocol leaks excess information . We show that our protocol offers better privacy and that it is simpler and is significantly more efficient in terms of communication rounds, communication cost and computational cost. In Existing System, the problem of secure mining of association rules in horizontally partitioned databases. In that setting, there are several sites that hold homogeneous databases, i.e., databases that share the same schema but hold information on different entities. 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. It was difficult to get accurate item set.

III. RESEARCH COMPONENT

The proposed protocol is for secure mining of association rules in horizontally distributed databases that improves significantly upon the current leading protocol in terms of privacy and efficiency. One of the main ingredients in our proposed protocol is a novel secure multi-party protocol for computing the union (or intersection) of private subsets that each of the interacting players hold. We propose a protocol for secure mining of association rules in horizontally distributed databases. The current leading protocol is that of Kantarcioglu and Clifton . Our protocol, like theirs, is based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [5], which is an unsecured distributed version of the Apriori algorithm. The main ingredients in our 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 site in a subset held by another. Our protocol offers enhanced privacy. It is simpler and is significantly more efficient in terms of communication rounds, communication cost and computational cost.

Here we propose an alternative protocol for the secure computation of the union of private subsets. The proposed protocol improves upon simplicity and efficiency as well as privacy. In particular, our protocol does not depend on commutative encryption and oblivious transfer. While our solution is still not perfectly secure, it leaks excess information only to a small number of possible coalitions, unlike the previous protocol that discloses information also to some single sites. In addition, we claim that the excess information that our protocol may leak is less sensitive than the excess information leaked by the already existing protocol.

IV. SCOPE OF PROBLEM

The protocol that we propose here computes a parameterized family of functions, which we call threshold functions, in which the two extreme cases correspond to the problems of computing the union and intersection of private subsets. 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 that our protocol may leak is less sensitive than the excess information leaked by other protocol.

A SECURE MULTIPARTY PROTOCOL FOR COMPUTING THE OR OF PRIVATE BINARY VECTORS
Protocol 2 (THRESHOLD) Secure computation of the *t* threshold function
Input: Each player P_m has an input binary vector $\mathbf{b}_m \in Z_n 2$, $1 \le m \le M$.
Output: $\mathbf{b} := Tt(\mathbf{b}1, \ldots, \mathbf{b}M)$.
1: Each P_m selects M random share vectors $\mathbf{b}_{m,\ell} \in Z_n$ $M+1$, $1 \le \ell \le M$, such that $\Sigma M\ \ell=1\ \mathbf{b}_{m,\ell} = \mathbf{b}_m \bmod (M + 1)$.
2: Each P_m sends $\mathbf{b}_{m,\ell}$ to P_ℓ for all $1 \le \ell \ne m \le M$.
3: Each $P\Sigma\ \ell$ computes $\mathbf{s}_\ell = (s_\ell(1), \ldots, s_\ell(n)) := M\ \ m=1\ \mathbf{b}_{m,\ell} \bmod (M + 1)$.
4: Players P_ℓ, $2 \le \ell \le M - 1$, send $\mathbf{s}_\ell$ to $P1$.
5: $\Sigma P1$ computes $\mathbf{s} = (s(1), \ldots, s(n)) := M-1\ \ \ell=1\ \mathbf{s}_\ell \bmod (M + 1)$.
6: for $i = 1, \ldots, n$ do
7: If $(s(i) + sM(i)) \bmod (M + 1) < t$ set $b(i) = 0$ otherwise set $b(i) = 1$.
8: end for
9: Output $\mathbf{b} = (b(1), \ldots, b(n))$.

V. THE FAST DISTRIBUTED MINING ALGORITHM

The protocol based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [5] ,which is an unsecured distributed version of the Apriori algorithm. Its main idea is that any *s*-frequent itemset must be also locally *s*-frequent in at least one of the sites. Hence, in order to find all globally *s*-frequent itemsets, each player reveals his locally *s*-frequent itemsets and then the players check each of them to see if they are *s*-frequent also globally.

The FDM algorithm proceeds as follows:
(1) Initialization: It is assumed that the players have already jointly calculated $F^{k-1}_{s.}$ The goal is to proceed and calculate $F^{k}_{s.}$
(2) Candidate Sets Generation: Each player P_m computes the set of all $(k-1)$ - itemsets that are locally frequent in his site and also globally frequent; namely, P_m computes the set
$F_s^{k-1,m} \cap F_s^{k-1}$. He then applies on that set the Apriori algorithm in order to generate the set
$B_s^{k,m}$ of candidate *k*-itemsets.
(3) Local Pruning: For each $X \in B_s^{k,m}$, Pm computes suppm(X). He then retains only those itemsets that are locally s-frequent. We denote this collection of itemsets by $C_s^{k,m}$.
(4) Unifying the candidate itemsets: Each player broadcasts his $C_s^{k,m}$ and then all players compute $C_s^k := U^M_{m=1} C_s^{k,m}$.
(5) Computing local supports. All players compute the local supports of all itemsets in C_s^k.
(6) Broadcast Mining Results: Each player broadcasts the local supports that he computed.

From that, everyone can compute the global support of every itemset in C_s^k. Finally, F^k_s is the subset of C_s^k that consists of all globally sfrequent k-itemsets.

The complete FDM algorithm starts by finding all single items that are globally *s*-frequent. It then proceeds to find all 2-itemsets that are globally *s*-frequent, and so forth, until it finds the longest globally *s*-frequent itemsets. If the length of such itemsets is *K*, then in the $(K+1)$th iteration of the FDM it will find no $(K+1)$-itemsets that are globally *s*-frequent, in which case it terminates.

VI. 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.In our problem, the inputs are the partial databases. As the inputs are the partial databases, the required output is the list of association rules that hold in the unified database with support and confidence no smaller that our protocol may leak is less sensitive than the excess information leaked by other protocol. The protocol is based on the Fast Distributed Mining algorithm. Our protocol may leak is less sensitive than the excess information leaked by other protocol. This new protocol offers enhanced privacy with respect to the previous protocol. In addition, it is simpler and is significantly more efficient in terms of communication rounds, communication cost and computational cost.

VII. PROTOCOL UNIFI-KC

Protocol UNIFI-KC works as follows-
First, each player adds to his private subset $C^{k,m}_s$ fake itemsets, in order to hide its size. Then, the players jointly compute the encryption of their private subsets by applying on those subsets a commutative encryption1, where each player adds, in his turn, his own layer of encryption using his private secret key. At the end of that stage, every itemset in each subset is encrypted by all of the players; the usage of a commutative encryption scheme ensures that all itemsets are, eventually, encrypted in the same manner. Then, they compute the union of those subsets in their encrypted form. Finally, they decrypt the union set and remove from it itemsets which are identified as fake.

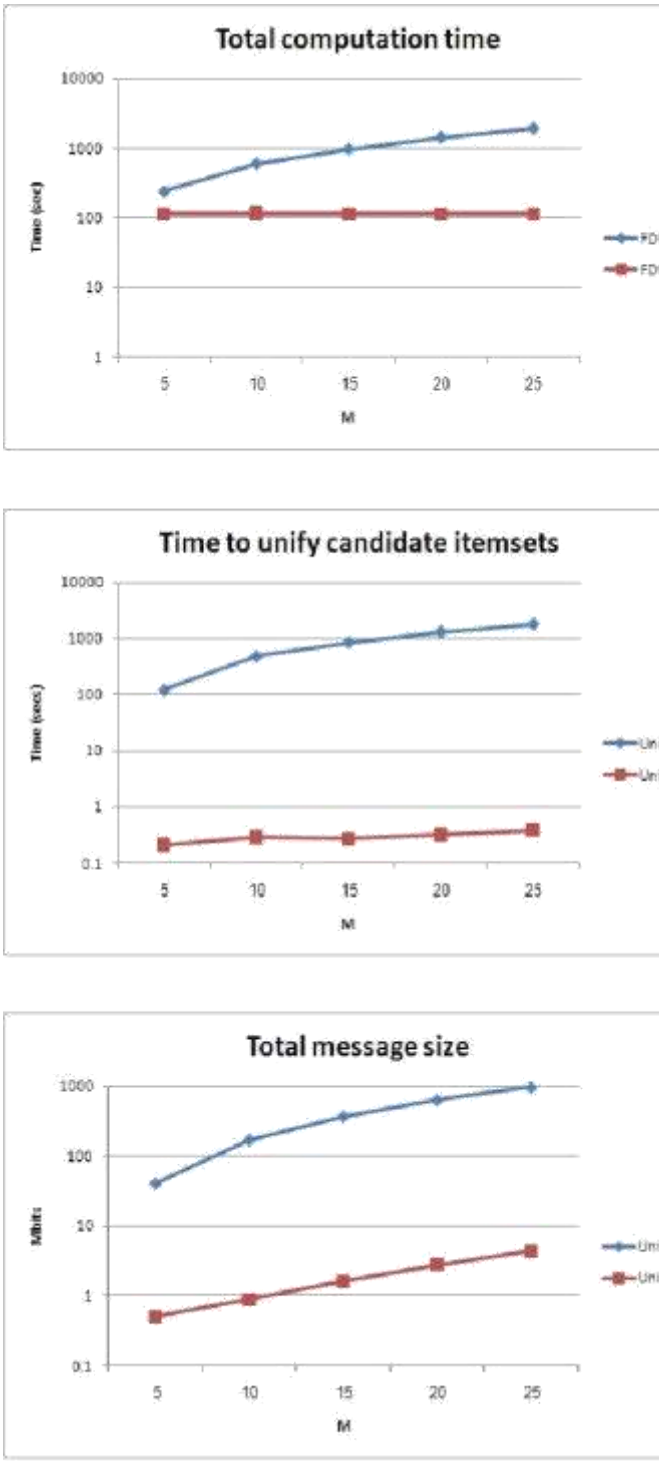

Figure 1: Computation and communication costs versus the number of players *M*

It is a pleasure to acknowledge the assistance of several people and institutions in this effort. First and foremost, I feel indebted to my guide, Professor R.K. Krishna, Department of Electronics, Rajiv Gandhi College of Engineering, Research &

Technology , Chandrapur for his valuable guidance, continuous support ,advice and constant encouragement throughout my work. A special word of thanks goes to Prof. Rahila Sheikh, Assistant professor, Department of Computer Technology and Prof. P.S.Kulkarni, Head, Department of Information Technology, R.C.E.R.T., Chandrapur for their encouragement to accomplish my work on time. I am also grateful to Prof. Nitin J. Janwe, Head, Department of Computer Technology, R.C.E.R.T., Chandrapur for his last minute instruction which helped me to focus my work in the right direction. I would like to extend my gratitude to honorable Dr.K.R.Dixit, Principal, R.C.E.R.T., Chandrapur, for being a constant source of inspiration.

Finally, I would like to extend my thanks to all those who have contributed directly or indirectly to make this project successful.

REFERENCES

[1] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules in large databases." In *VLDB*, pages 487–499, 1994.

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

[3] J.C. Benaloh. "Secret sharing homomorphisms: Keeping shares of a secret." In *Crypto*, pages 251–260, 1986.

[4] J. Brickell and V. Shmatikov." Privacy-preserving graph algorithms in the semi-honest model." In *ASIACRYPT*, pages 236–252, 2005.

[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] D.W.L Cheung, V.T.Y. Ng, A.W.C. Fu, and Y. Fu." Efficient mining of association rules in distributed databases. "*IEEE Trans. Knowl. Data Eng.*, 8(6):911–922, 1996.

[7] T. ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms." *IEEE Transactions on Information Theory*, 31:469 472, 1985.

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

[9] R. Fagin, M. Naor, and P. Winkler. "Comparing Information Without Leaking It."*Communications of the ACM*, 39:77–85, 1996.

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

[11] M.J. Freedman, K. Nissim, and B. Pinkas." Efficient private matching and set intersection. "In *EUROCRYPT*, pages 1–19, 2004.

[12] 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.

[13] M. Bellare, R. Canetti, and H. Krawczyk. "Keying hash functions for message authentication. "In *Crypto*, pages 1–15, 1996.

[14] A. Ben-David, N. Nisan, and B. Pinkas. FairplayMP – "A system for secure multi-party computation." In *CCS*, pages 257–266, 2008.

[15] D. Beaver, S. Micali, and P. Rogaway. "The round complexity of secure protocols. In *STOC"*, pages 503–513, 1990.

[16] 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.