NOVEL SECURE MULTIPARTY PROTOCOL IN DISTRIBUTED DATABASES

P.LALITHA, P.RAMESWARA ANAND**, P.NAGESWARA RAO****

**M.Tech Student of Swetha Institute of Technology and Science, Tirupathi*

*** Associate Professor, Dept. of CSE, Swetha Institute of Technology and Science, Tirupathi*

****Head Dept. of CSE, Swetha Institute of Technology and Science, Tirupathi*

Abstract- We propose a protocol for secure mining of association rules in horizontally distributed databases. Our protocol, like theirs, is based on the Fast Distributed Mining (FDM) algorithm, 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 player in a subset held by another. In addition, it is simpler and is significantly more efficient in terms of communication rounds, communication cost and computational cost.

Index Terms- Privacy Preserving Data Mining; Distributed Computation; Frequent Item sets; Association Rules.

I. INTRODUCTION

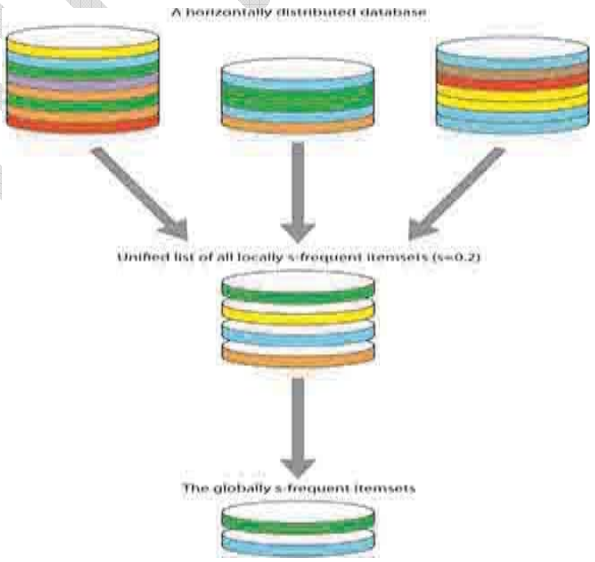
We study here the problem of secure mining of association rules in horizontally partitioned databases. In that setting, there are several sites (or players) that hold homogeneous databases, i. e., databases that share the same schema but hold information on different entities. The goal is to find all association rules with support at least s and confidence at least c, for some given minimal support size s and confidence level c, that hold in the unified database, while minimizing the information disclosed about the private databases held by those players .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, x1. . . xM, and they wish to securely compute y = f (x1. . . xM) 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 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. As the abovementioned generic solutions rely upon a description of the Function f as a Boolean circuit, they can be applied only to small inputs and functions which are realizable by simple circuits. In more complex settings, such as ours, other methods are required for carrying out this computation.

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 he item sets that are s-frequent in his partial database.) Than 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. Herein we propose an alternative protocol for the secure computation of the union of private subsets. The proposed protocol improves upon that in terms of simplicity and efficiency as well as privacy. In particular, our protocol does not depend on commutative encryption and oblivious transfer (what simplifies it significantly and contributes towards much reduced communication and computational costs). While our solution is still not perfectly secure, it leaks excess information only to a small number (three) of possible coalitions, unlike the protocol of that discloses information also to some single players. 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. Those are in fact general-purpose protocols that can be used in other contexts as well. Another problem of secure multiparty computation that we solve here as part of our discussions the set inclusion problem; namely, the problem where Alice holds a private subset of some ground set, and Bob holds an element in the ground set, and they wish to determine whether Bob's element is within Alice's subset, without revealing to either of them information about the other party's input beyond the above described inclusion.

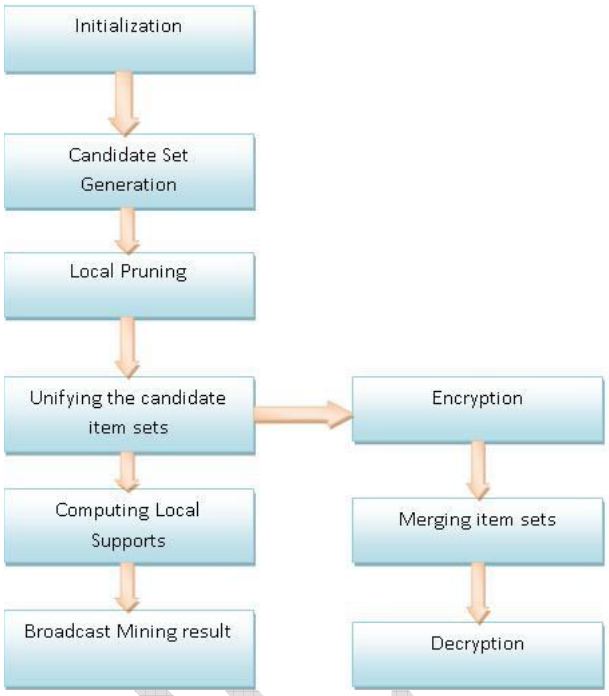
Overview and Organization of the Paper: Where the players broadcast the item sets that are locally frequent in their private databases, and in Step 6, where they broadcast the sizes of the local supports of candidate item sets. Our improvement is with regard to the secure implementation of Step 4, which is the more costly stage of the protocol, and the one in which the protocol of leaks excess information. We then describe our alternative implementation and proceed to analyze the two implementations in terms of privacy and efficiency and compare them. 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. We discuss the implementation of the two remaining steps of the distributed protocol: The identification of those candidate item sets that are globally s frequent, and then the derivation of all (s, c)-association rules.



II. SYSTEM ARCHITECTURE

System architecture describes the flow of data inside the system. It goes through various phases as shown in figure number 3. It is having initialization, in which the player is starting their role by holding some value in it. And then it will help to find out the next item. Next phase is generating candidate set, in which we are finding the key which appears repeatedly or you may say it which is intersection or common for both sites and

players. Next phase is local pruning, in which we are trying to eliminate the unwanted result or extra data which will in turn help in mining the data. Next phase is Candidate key union, as word indicates it is based on the union of data of participating players. Next phase is local support computation, in which we are computing the local support that how much the participating player can support. Next phase is Broadcasting of the mining result in which we are going to display the result by merging the all result that we got from all participating player and then displaying it.



System Architecture

III. SECURE COMPUTATION OF ALL LOCALLY FREQUENT ITEM SETS

Here we discuss the secure implementation of Step 4 in the FDM algorithm, namely, the secure computation of the union Cks =∪M m=1 Ck,ms . We describe the protocol of and then our protocol. We analyze the privacy of the two protocols their communication cost in and their computational cost.

3.1 The protocol of Kantarcioglu and Clifton for the secure computation of all locally frequent item sets:

Protocol 1 is the protocol that was suggested by Kantarcioglu and Clifton for computing the unified list of all locally frequent itemsets, Cks =∪M m=1 Ck, ms, without disclosing the sizes of the subsets Ck, ms nor their contents. The protocol is applied when the players already know Fk−1s — the set of all(k−1)-itemsets that are globally s-frequent, and they wish to proceed and compute Fks . We refer to it hereinafter as Protocol UNIFI-KC (Unifying lists of locally Frequent Itemsets —The input that each player Pm has at the beginning of Protocol UNIFI-KC is the collection Ck,ms , as defined in Steps2-3 of the FDM algorithm. Let Ap (Fk−1 s) denote the set of all candidate k-itemsets that the Apriori algorithm generates fromFk−1s . Then, as implied by the definition of Ck, ms (see Section1.1.2), Ck, ms , $1 \leq m \leq$ M, are all subsets of Ap(Fk−1s). The output of the protocol is the union Cks =∪Mm=1 Ck,ms . In the first iteration of this computation k = 1, and the players compute all s-frequent 1-itemsets (here F0s = {∅}). In the next iteration they compute all s-frequent 2-itemsets, and so forth, until the first $k \leq L$ in which they find no s-frequent k-itemsets. After computing that union, the players proceed to extract from Cks the subset Fks that consists of all k-itemsets that are globally s-frequent; this is done using the protocol that we describe later on in Section 3. Finally, by applying the above described procedure from k = 1 until the first value of $k \leq L$ for which the resulting set Fks is empty, the players may recover the full set Fs :=∪Lk=1 Fks of all globally s-frequent itemsets. Protocol UNIFI-KC works as follows: First, each player adds to his private subset Ck,ms 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. We now proceed to describe the protocol in detail.

IV. IDENTIFYING THE GLOBALLY S-FREQUENT ITEM SETS

Protocols UNIFI-KC and UNIFI yield the set Cks that consists of all itemsets that are locally s-frequent in at least one site. Those are the k-itemsets that have potential to be also globally s-frequent. In order to reveal which of those itemsets is globally s-frequent there is a need to securely compute the support of each of those itemsets. That computation must not reveal the local support in any of the sites. Let x be one of the candidate itemsets in Cks . Then x is globally s-frequent if and only if $\Delta(x) := supp(x) - sN = M\Sigma m=1 (suppm(x) - sNm) \geq 0$. (7). we describe here the solution that was proposed by Kantarcioglu and Clifton. They considered two possible settings. If the required output includes all globally s-frequent item sets,as well as the sizes of their supports, then the values of $\Delta(x)$ can be revealed for all xU Cks . In such a case, those values may be computed using a secure summation protocol, where the private addend of Pm is $suppm(x) - sNm$. The more interesting setting, however, is the one where the support sizes are not part of the required output. We proceed to discuss it. As $|\Delta(x)| \leq N$, an item set x U Cks is s-frequent if and only if $\Delta(x) \bmod q \leq N$, for $q = 2N + 1$. The idea is to verify that inequality by starting an implementation of the secure summation protocol of [6] on the private inputs $\Delta m(x) := suppm(x) - sNm$, modulo q. In that protocol, all players jointly compute random additive shares of the required sum $\Delta(x)$ and then, by sending all shares to, say, P1, he may add them and reveal the sum. If, however, PM withholds his share of the sum, then P1 will have one random share, s1(x), of $\Delta(x)$, and PM will have a corresponding share, sM(x); namely, $s1(x)+sM(x) = \Delta(x) \bmod q$. It is then proposed that the two players execute the generic secure circuit evaluation of [32] in order to verify whether $(s1(x) + sM(x)) \bmod q \leq N$.

Those circuit evaluations may be parallelized for all x U Cks. We observe that inequality holds if and only if $s1(x) U \Theta(x) := \{(j - sM(x)) \bmod q : 0 \leq j \leq N\}$ As s1(x) is known only to P1 while $\Theta(x)$ is known only to PM, the verification of the set inclusion in (9) can also be carried out by means of Protocol SETINC. However, the ground set Ω in this case is Zq=2N+1, which is typically a large set. (Recall that when Protocol SETINC is invoked from UNIFI, the ground set Ω is ZM+1, which is usually a small set.) Hence, Protocol SETINC is not useful in this case, and, Consequently, Yao's generic protocol remains, for the moment, the protocol of choice to securely verify inequality. Yao's protocol is designed for the two-party case. In our setting, as M > 2, there exist additional semi-honest players. An interesting question which arises in this context is whether the existence of such additional semi-honest players may be used to verify inequalities like (8), even when the modulus is large, without resorting to costly protocols such as oblivious transfer.

V. IDENTIFYING ALL(S, C) ASSOCIATION RULES

Once the set Fs of all s-frequent item sets is found, we may proceed to look for all (s, c)-association rules (rules with support at least sN and confidence at least c), as described in [18]. For X, Y ∈ Fs, where $X \cap Y = \emptyset$, the corresponding association rule X U Y has confidence at least c if and only if $supp(X \cup Y)/supp(X) \geq c$, or, equivalently, $CX,Y := M\Sigma m=1 (suppm(X \cup Y) - c \bullet$

suppm(X)) ≥ 0 . (10)Since |CX,Y | ≤ N, then by taking q = 2N+1, the players can verify inequality (10), in parallel, for all candidate association rules, as described in Section 3.In order to derive from Fs all (s, c)-association rules in an efficient manner we rely upon the following straightforward lemma.

VI. RELATED WORK

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 anonym zing the data prior to its release. The main approach in this context is to apply data perturbation. The idea is that Fig. 1. Computation and communication costs versus the number of transactions N 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 usual approach here is cryptographic rather than probabilistic. Discussed secure clustering using the EM algorithm over horizontally distributed data. The problem of distributed association rule mining was studied in the vertical setting, where each party holds a different set of attributes, and in [18] in the horizontal setting. Also the work of considered this problem in the horizontal setting, but they considered large-scale systems in which, on top of the parties that hold the data records (resources) there are also managers which Fig. 2. Computation and communication costs versus the number of players M are computers that assist the resources to decrypt messages; another assumption made in that distinguishes it from [18] and the present study is that no collusions occur between the different network nodes — resources or managers.

VII. CONCLUSION

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. Another ingredient is a protocol that tests the inclusion of an element held by one player in a subset held by another. Those protocols exploit the fact that the underlying problem is of interest only when the number of players is greater than two. We proposed a protocol for secure mining of association rules in horizontally distributed databases that improves significantly upon the current leading protocol in terms of privacy and efficiency. namely, to devise an efficient protocol for inequality verifications that uses the existence of a semi honest third party. Such a protocol might enable to further improve upon the communication and computational costs of the second and third stages of the protocol of. Other research problems that this study suggests is the implementation of the techniques presented here to the problem of distributed association rule mining in the vertical setting problem of mining generalized association rules and the problem of subgroup discovery in horizontally partitioned data.

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] 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] A. Ben-David, N. Nisan, and B. Pinkas. Fairplay MP - A system for secure multi-party computation. In CCS, pages 257–266, 2008.

[6] J.C. Benaloh. Secret sharing homeomorphisms: Keeping shares of a secret Secret. In Crypto, pages 251–260, 1986.

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

BIOGRAPHY

Author Details: P.LALITHA Student of M. Tech., Swetha Institute of Technology and Science, Tirupathi

Email: dhatri.sai18@gmail.com

Guide Details: P.RAMESWARA ANAND, Associate Professor, Dept. of CSE, Swetha Institute of Technology and Science, Tirupathi

Guide Details: P. NAGESWARA RAO, Head Dept. of CSE, Swetha Institute of Technology and Science, Tirupathi