Secure Mining of Association Rulues over Horizontally Partioned Data in Data Mining

M.Mohan Rao ,Asst.professor[1], M.Laxmaiah[2]

GJCST Classification H.2.8

***Abstract*-Data mining can extract important knowledge from large data collection. Sometimes these large collections are split among various parties, which can share the data. For example, insurance companies share the data from medical hospitals. Privacy concerns may prevent the parties from directly sharing the data. Here this project addresses secure mining of association rules over horizontally partitioned data. The existing work provides security by using some techniques like Randomization, Secure Multi Party computation, etc., The drawbacks in the existing work are inaccuracy, inefficiency and lacking of security. To overcome the drawbacks of existing system we proposed a new method by using Commutative Encryption tool, Randomization and Secure multi party computation. This work also provides security for multiple numbers of sites. Examples include knowledge discovery among intelligence services of different countries and collaboration among corporations without revealing trade secrets.**

Keywords-Data Mining, Security, 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 custodians, none of which are allowed to transfer their data to another site.This paper addresses the problem of computing association rules.The goal is to produce association rules that hold globally while limiting the information shared about each site. We can compute the global support and confidence of an association rule AB=>C knowing only the local supports of AB and ABC and size of each database.

$\text{support}_{AB=>C} = \sum_{i=1}^{\text{sites}} \text{support count}_{ABC}(i) / \sum_{i=1}^{\text{sites}} \text{database\_size}(i)$

$\text{support}_{AB} = \sum_{i=1}^{\text{sites}} \text{support count}_{AB}(i) / \sum_{=1}^{\text{sites}} \text{database size}(i)$

$\text{confidence}_{AB=>C} = \text{support}_{AB=>C} / \text{support}_{AB}$

1) *Private Association Rule Mining Overview:*

Our method follows the two-phase approach described above, but combining locally generated rules and support counts is done by passing encrypted values between sites. The two phases are discovering candidate itemsets (those that are frequent on one or more sites) and determining which of the candidate itemsets meet the global support/ confidence thresholds.The first phase (Fig. 1) uses commutative encryption. Each party encrypts its own frequent itemsets (e.g., Site 1 encrypts itemset C). The encrypted itemsets are then passed to other parties until all parties have encrypted all itemsets. These are passed to a common party to eliminate duplicates and to begin decryption. (In the figure, the full set of itemsets are shown to the left of Site 1, after Site 1 decrypts.) This set is then passed to each party and each party decrypts each itemset. The final result is the common itemsets (C and D in the figure).In the second phase (Fig. 2), each of the locally supported itemsets is tested to see if it is supported globally. In the figure, the itemset ABC is known to be supported at one or more sites and each computes their local support. The first site chooses a random value R and adds to R the amount by which its support for ABC exceeds the minimum support threshold. This value is passed to site 2, which adds the amount by which its support exceeds the threshold (note that this may be negative, as shown in the figure.) This is passed to site 3, which again adds its excess support. The resulting value (18) is tested using a secure comparison to see if it exceeds the Random value (17). If so, itemset ABC is supported globally.

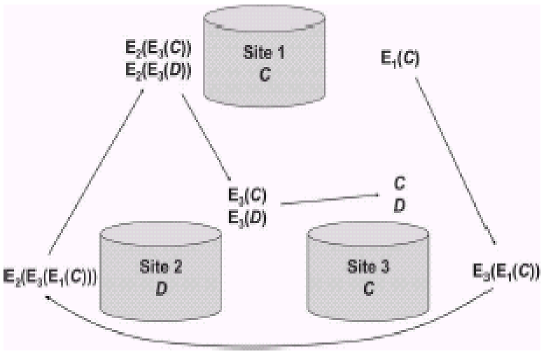

Fig. 1. Determining global candidate itemsets

About[1]- Tirumala engg college, ***Mohanrao19@yahoo.com***
About[2]- Associate Professor & Hod, Tirumala engg College , laxman_mettu@yahoo.com.

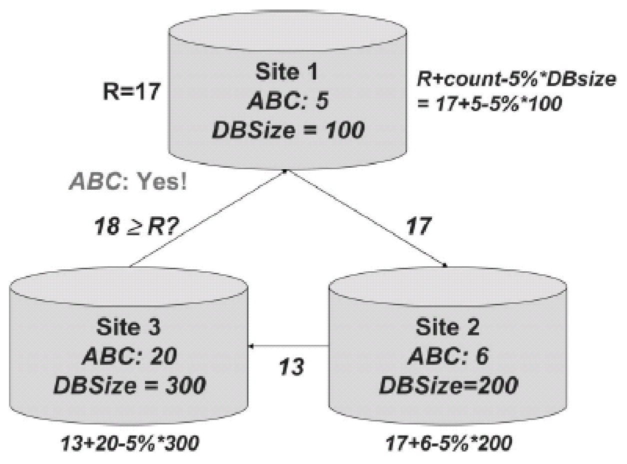

Fig. 2. Determining if item set support exceeds 5 percent threshold

II. Background and Related Work

There are several fields where related work is occurring. We first describe other work in privacy-preserving data mining, then go into detail on specific background work on which this paper builds._Previous work in privacy-preserving data mining has addressed two issues. In one, the aim is preserving customer privacy by distorting the data values [2]. The idea is that the distorted data does not reveal private information and thus is safe" to use for mining. More recently, the data distortion approach has been applied to Boolean association rules. Again, the idea is to modify data values such that reconstruction of the values for any individual transaction is difficult, but the rules learned on the distorted data are still valid. One interesting feature of this work is a flexible definition of privacy, e.g., the ability to correctly guess a value of "1" from the distorted data can be considered a greater threat to privacy than correctly learning a 0."The other approach uses cryptographic tools to build decision trees. In this work, the goal is to securely build an ID3 decision tree where the training set is distributed between two parties. The basic idea is that finding the attribute that maximizes information gain is equivalent to finding the attribute that minimizes the conditional entropy. The conditional entropy for an attribute for two parties can be written as a sum of the expression of the form (v1+ v2)*log(v1+v2). The authors give a way to securely calculate the expression (v1+v2)*log(v1+v2) and show how to use this function for building the ID3 securely.

1) *Mining of Association Rules*

The association rules mining problem can be defined as follows [1]: Let I ={ i1,i2, . . .,in} be a set of items. Let DB be a set of transactions where each transaction T is an itemset such that $T \subseteq I$. Given an itemset $X \subseteq I$, a transaction T contains X if and only if $X \subseteq T$. An association rule is an implication of the form X =>Y, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \Phi$. The rule X =>Y has support s in the transaction database DB if s% of transactions in DB contains $X \cup Y$. The association rule holds in the transaction database DB with confidence c if c% of transactions in DB that contain X also contains Y. An itemset X with k items is called k-itemset. The problem of mining association rules is to find all rules whose support and confidence are higher than certain user-specified minimum support and confidence. In this simplified definition of the association rules, missing items, negatives, and quantities are not considered. In this respect, transaction database DB can be seen as 0/1 matrix where each column is an item and each row is a transaction. In this paper, we use this view of association rules.

2) *Distributed Mining of Association Rules:*

The above problem of mining association rules can be extended to distributed environments. Let us assume that a transaction database DB is horizontally partitioned among n sites (namely, S1,S2, . . . , Sn) where DB = DB1$\cup$ DB2 $\cup$. . .$\cup$ DBn and DB_i resides at side S_i(1<=I<=n). The itemset X has local support count of X.sup_i at site S_I if X.sup_I of the transactions contains X. The global support count of X is given as X.sup = $\sum^{N}_{I=1}$ X.sup_i. An itemset X is globally supported if X.sup>=s*$\sum^{N}_{I=1}|DB_i|$. Global confidence of a rule X =>Y can be given as {X $\cup$Y}.sup/X.sup.A fast algorithm for distributed association rule mining is given in Cheung et al. [1]. Their procedure for fast distributed mining of association rules (FDM) is summarized below:

1) Candidate Sets Generation: Generate candidate sets $CG_{I(k)}$ based on $GL_{I(k-1)}$, itemsets that are supported by the S_i at the (k-1)th iteration, using the classic a priori candidate generation algorithm. Each site generates candidates based on the intersection of globally large (k- 1) itemsets and locally large (k- 1) itemsets.

2) Local Pruning: For each X $\Xi CG_{I(k)}$, scan the database DBi at Si to compute X:supi. If X is locally large S_i, it is included in the LLi set. It is clear that if X is supported globally, it will be supported in one site.

3) Support Count Exchange: $LL_{i(k)}$are broadcast and each site computes the local support for the items in $ULL_{i(k)}$.

4) Broadcast Mining Results*:* Each site broadcasts the local support for itemsets in $ULL_{i(k)}$. From this, each site is able to compute $L_{(k)}$.

3) *Secure Multiparty Computation:*

Substantial work has been done on secure multiparty computation. The key result is that a wide class of computations can be computed securely under reasonable assumptions. We give a brief overview of this work, concentrating on material that is used later in the paper. The definitions given here are from Goldreich. For simplicity, we concentrate on the two-party case. Extending the definitions to the multiparty case is straightforward.

1) Security in Semihonest Model:

A semihonest party follows the rules of the protocol using its correct input, but is free to later use what it sees during execution of the protocol to compromise security. This is somewhat realistic in the real world because parties who want to mine data for their mutual benefit will follow the protocol to get correct results. Also, a protocol that is buried in large, complex software cannot be easily altered. A formal definition of private two-party computation in the semihonest model is defined. Computing a function privately is equivalent to computing it securely.

2) Yao's General Two-Party Secure Function Evaluation

Yao‘s general secure two-party evaluation is based on expressing the function f(x,y) as a circuit and encrypting the gates for secure evaluation [3]. With this protocol, any two-party function can be evaluated securely in the semihonest model. To be efficiently evaluated, however, the functions must have a small circuit representation. We will not give details of this generic method; however, we do use this generic result for securely finding whether a >=b (Yao‘s millionaire problem). For comparing any two integers securely, Yao‘s generic method is one of the most efficient methods known, although other asymptotically equivalent but
practically more efficient algorithms could be used as well.

3) Commutative Encryption

Commutative encryption is an important tool that can be used in many privacy-preserving protocols. An encryption algorithm is commutative if the following two equations hold for any given feasible encryption keys k1, k2,... ., Kn ∊ K, any message M, and any permutations of i,j:

$$E_{ki1}(\ldots E_{kin}(M)\ldots..)= E_{ji1}(\ldots E_{jin}(M)\ldots..)$$

$\forall$ M1,M2 ∊ M such that M1≠ M2 and for given k, $\in<1/2^k$

$$Pr(E_{ki1}(\ldots E_{kin}(M)\ldots..)= E_{ji1}(\ldots E_{jin}(M)\ldots..))< \in$$

These properties of commutative encryption can be used to check whether two items are equal without revealing them. For example, assume that party A has item i_A and party B has item i_B. To check if the items are equal, each party encrypts its item and sends it to the other party: In addition to meeting the above requirements, we require that the encryption be secure. Specifically, the encrypted values of a set of items should reveal no information about the items themselves.

III. SECURE ASSOCIATION RULE MINING

We will now use the tools described above to construct a distributed association rule mining algorithm that preserves the privacy of individual site results. The algorithm given is for three or more parties.

4) *Problem Definition*

Let i >=3 be the number of sites. Each site has a private transaction database DBi. We are given support threshold s and confidence c as percentages. The goal is to discover all association rules satisfying the thresholds, as defined in Section 2.1.1. We further desire that disclosure be limited: No site should be able to learn contents of a transaction at any other site, what rules are supported by any other site, or the specific value of support/confidence for any rule at any other site unless that information is revealed by knowledge of one‘s own data and the final result. For example, if a rule is supported globally but not at one‘s own site, we can deduce that at least one other site supports the rule. Here, we assume no collusion.

5) *Method*

Our method follows the general approach of the FDM algorithm, with special protocols replacing the broadcasts of $LL_{i(k)}$ and the support count of items $LL_{i(k)}$. We first give a method for finding the union of locally supported itemsets without revealing the originator of the particular itemset. We then provide a method for securely testing if the support count exceeds the threshold.

6) *Secure Union of Locally Large Itemsets*

In the FDM algorithm (Section 2.1.1), Step 3 reveals the large item sets supported by each site. To accomplish this without revealing what each site supports, we instead exchange locally large itemsets in a way that obscures the source of each itemset. We assume a secure commutative encryption algorithm with negligible collision probability The main idea is that each site encrypts the locally supported itemsets, along with enough fake" itemsets to hide the actual number supported. Each site then encrypts the itemsets from other sites. In Phases 2 and 3, the sets of encrypted itemsets are merged. Since (3) holds, duplicates in the locally supported itemsets will be duplicates in the encrypted itemsets and can be deleted. The reason this occurs in two phases is that if a site knows which fully encrypted itemsets come from which sites, it can compute the size of the intersection between any set of sites. While generally innocuous, if it has this information for itself, it can guess at the itemsets supported by other sites. Permuting the order after encryption in Phase 1 prevents knowing exactly which itemsets match; however, separately merging itemsets from odd and even sites in Phase 2 prevents any site from knowing the fully encrypted values of its own itemsets. Phase 4 decrypts the merged frequent itemsets. Commutativity of encryption allows us to decrypt all itemsets in the same order regardless of the order they were encrypted in, preventing sites from tracking the source of each itemset. The detailed algorithm is given in Protocol 1 (see Fig. 3).

IV. CONCLUSIONS

Naturally, ever increasing data collection, along with the influx of analysis tools capable of handling huge volumes of information, has led to privacy concerns. The cryptographic community has formally defined much stronger notions of privacy. Also cryptography tools can enable data mining that would otherwise be prevented due to security concerns. We have given procedures to mine distributed association rules on horizontally partitioned data. We have shown that distributed association rule mining can be done efficiently under reasonable security assumptions. We believe the need for mining of data where access is restricted by privacy concerns will increase. Examples include knowledge discovery among intelligence services of different countries and collaboration among corporations without revealing trade secrets. Even within a single multinational company, privacy laws in different jurisdictions may prevent sharing individual data. Allowing error in the results may enable more efficient algorithms that maintain the desired level of security. Continued research will expand the scope of privacy-preserving data mining, enabling most or all data mining methods applied in situations where privacy concerns would appear to restrict such mining.

```
Protocol 1 Finding secure union of large itemsets of size k
Require: N= 3 sites numbered 1…..N1, set F of nonitemsets
Phase 0: Encryption of all the rules by all sites
for each site i do
generate LLi(k) as in steps 1 and 2 of the FDM algorithm
LLei(k) = ø;
for each X € LLi(k) do
LLei(k) = LLei(k) U{Ei(X)}
end for
for j = |LLei(k)| + 1 to |CG(k)| do
LLei(k) = LLei(k) [ {Ei(random selection from F)}
end for
end for
Phase 1: Encryption by all sites
for Round j = 0 to N − 1 do
if Round j= 0 then
Each site i sends permuted LLei(k) to site (i+1) mod N
else
Each site i encrypts all items in LLe(i−j mod N)(k) with Ei, permutes, and sends it to site (i+1) mod N
end if
end for{At the end of Phase 1, site i has the itemsets of site (i + 1) mod N encrypted by every site}
Phase 2: Merge odd/even itemsets
Each site i sends LLei+1 mod N to site i mod 2
Site 0 sets RuleSet1 = [d(N−1)/2e
j=1 LLe(2j−1)(k)
Site 1 sets RuleSet0 = [b(N−1)/2c
j=0 LLe(2j)(k)
Phase 3: Merge all itemsets
Site 1 sends permuted RuleSet1 to site 0
Site 0 sets RuleSet = RuleSet0 U RuleSet1
Phase 4: Decryption
for i = 0 to N − 1 do
Site i decrypts items in RuleSet using Di
Site i sends permuted RuleSet to site i + 1 mod N
end for
Site N 1 decrypts items in RuleSet using DN1
RuleSet(k) = RuleSet F
Site N 1 broadcasts RuleSet(k) to sites 0..N 2
```

Fig. 3. Protocol 1: Finding secure union of large itemsets of size k.

3.3 Algorithm to Calculate Global Support

```
Step 1 Calculate s for the data item whose support count is to be calculated
Step 2 Calculate $d$
Step 3 choose $r$ such that $0<r<d$
Step 4 calculate $a = d/k$
Step 5 Calculate $t= r+a$
Step 6 if $s>t$ then the data item is public
Else the data item is private
```

Thus the global support is calculated and the secret and the non secret data are thus separated.

V. REFERENCES

1) Achlioptas, D. (2004). Random matrices in data analysis. Proceedings of the 15th European Conference on Machine Learning, pp. 1-8, Pisa, Italy.
2) J. Vaidya and C. Clifton, Prviacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 639-644,2002,available: http://doi.acm.org/ 10.1145/775047.775142.
3) Wang, J., Zhang, J., Zhong, W. J., & Xu, S. (2007). A novel data distortion approach via selective SSVD for privacy protection. *International Journal of Information and Computer Security*, to appear.
4) R. Agrawal and R. Srikant, Privacy-Preserving Data Mining," Proc. 2000 ACM SIGMOD Conf. Management of Data, pp. 439-450, 2000, available: http://doi.acm.org/10.1145/342009.335438.
5) A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, Prviacy Preserving Mining of Association Rules," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 217-228,2002,available: http://doi.acm.org/ 10.1145/775047.775080.

6) R.L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978, available: http://doi.acm.org/ 10.1145/359340.359342.
7) S.C. Pohlig and M.E. Hellman, An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance," IEEE Trans. Information Theory, vol. IT-24, pp. 106-110, 1978.
8) O. Goldreich, Encryption Schemes," (working draft), Mar. 2003, available: http:// www.wisdom.eizmann.ac.il/~oded/PSBookFr ag/ enc.ps.
9) Y. Lindell and B. Pinkas, Privacy Preserving Data Mining," Advances in Cryptology (CRYPTO 2000), pp. 36-54, 2000, available: http:// link.springer.de/link/service/series/0558/bibs/ 1880/18800036.htm.
10) I. Ioannidis and A. Grama, An Efficient Protocol for Yao's Millionaires'Problem," Proc. Hawaii Int'l Conf. System Sciences (HICSS-36), 2003.
11) M.K. Reiter and A.D. Rubin, Crowds: Anonymity for Web Transactions," ACM Trans. Information

and System Security, vol. 1, no. 1, pp. 66-92, Nov. 1998, available: http://doi.acm.org/10.1145/290163.290168.

12) J.C. Benaloh, Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret," Advances in Cryptography (CRYPTO86): Proc., A. Odlyzko, ed., pp. 251-260, 1986, available:
13) B. Chor and E. Kushilevitz, A Communication-Privacy Tradeoff for Modular Addition," Information Processing Letters, vol. 45, no. 4, pp. 205-210, 1993.
14) C. Clifton, M. Kantarcioglu, and J. Vaidya, Defining Privacy for Data Mining," Proc. US Nat'l Science Foundation Workshop on Next Generation Data Mining, H. Kargupta, A. Joshi, and K. Sivakumar, eds., pp. 126-133, 2002.
15) T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Trans. Information Theory, vol. IT-31, no. 4, pp. 469-472, July 1985.
16) A. Shamir, R.L. Rivest, and L.M. Adleman, Mental Poker," Technical Memo MIT-LCS-TM-125, Laboratory for Computer Science, MIT, Fe