ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

A Novel Approach for Secure Mining of Association Rules with Multi-Party Protocol

Dr.P.Sumitra[1], S.Kokila[2]

Assistant Professor, Department of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam, Tiruchengode, Tamil Nadu, India

M.Phil Research Scholar, Department of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam, Tiruchengode, Tamil Nadu, India

ABSTRACT The research in data mining is developing fast and efficient algorithm to derive knowledge from huge databases. There are several data mining algorithms available to solve diverse data mining problems. They are mainly classified as Associations, Classifications, Sequential Patterns and Clustering. Apriori is one of the most important algorithms used in Rule Association Mining. 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., 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. The drawbacks of the existing system may produce a larger number of candidate item sets and scan the database many times. The proposed algorithm is based on the reverse scan of a given database. The proposed algorithm can greatly reduce the scanning times required for the discovery of candidate itemsets. Therefore, much time and space has been saved while searching frequent itemsets.

KEYWORDS: Data mining, Privacy Preserving Data Mining, Fast Distributed Mining Algorithm, Distributed Computation.

I.INTRODUCTION

Data mining techniques have been introduced successfully to retrieve knowledge in order to support a variety of domains marketing, weather forecasting, medical diagnosis, and national security. But it is still a challenge to mine the data by protecting the private database of user. Most organizations want information about individuals for their own specific needs. There is a common problem in many areas of science: obtaining adequate data. A particular frustration applies in some areas where the data exists, but is held in such a way that the data may not be accessed. Common examples arise in health science, where data may be held by multiple parties: commercial organizations (such as drug companies, or hospitals), government bodies (such as the Food and Drug Administration) and non-government organizations (such as charities). Each organization is bound by regulatory restrictions (for instance privacy legislation, e.g. [1]), and corporate requirements (for instance on distributing proprietary information that may provide commercial advantage to competitors). The same problem has arisen in analysis of the Internet. The data of interest concerns Internet properties of global interest: for instance, traffic, network performance, routing, and topology. These data are of obvious interest to Internet researchers, and there are also commercial applications that require collection of data from disparate sources

We study here the problem of secure mining of association rules in vertically partitioned databases. Association rule mining is an active data mining research area. However, most ARM algorithms cater to a centralized environment. Current technology for mining data typically applies to data stored centrally. A central part of many algorithms for mining association rules in large data sets is a procedure that is to find so called frequent item sets. The frequent item sets are very large due to transactions data increasing. The existing approach Fast Distributed Mining (FDM) algorithm is an unsecured distributed version of the Apriority algorithm.

The distributed mining algorithms can be used on distributed databases, as well as for mining large databases by partitioning them between sites and processing them in a distributed manner. The high flexibility, the scalability, the small cost/performance ratio and the connectivity of a distributed system make them an ideal platform for data mining. An association rule is a rule which implies certain association relationship among set of objects in a database. Since finding interesting association rules in a database may disclose some useful patterns for decision support, selective marketing, financial forecast, medical diagnosis and many other application it has attracted a lot of attention in recent data mining research. With the existence of many large transaction databases, huge amount of data, high scalability of distributed system, and the easy partition and distribution of a centralized database, it is important to investigate efficient methods for distributed mining of association rule. Association rule mining finds interesting association or correlation relationships among a large set of data items. Sensitive data is a part of every large organization's normal business practice.

II.LITERATURE SURVEY

The distributed association rule mining (ARM) problem has been studied for nearly a decade. Until recently, however, none of the algorithms presented for this problem could scale above several dozens of participants. The first scalable algorithm for the distributed ARM problem was presented in [12]. Since our algorithm is based on that work, we thoroughly describe it in Section 4. Nevertheless, the algorithm does not preserve privacy; specifically, if a resource communicates with the system via a single neighbor, then that neighbor will learn the resource’s statistics (e.g., the mortality rate for that HMO).

Privacy-preserving data mining has received a lot of attention in the past few years. In [3, 5], techniques based on perturbing the original data before initiating the mining process were used. However, this approach can only secure the privacy of records and not of the sources (e.g., of the patients but not of the HMOs).An alternative to the perturbation approach is to develop cryptographically secured versions of the data mining algorithm. This has been shown to be possible for three data mining problems: distributed ARM (the same problem discussed in this paper) [10], ARM in vertically partitioned data [11] – i.e., where each transaction is split among several nodes, and decision tree induction [9]. The main problem with these three algorithms is that the cryptographic primitives they use are global and rigid. The evaluation of every primitive requires the participation of all of the nodes, and if the data at even one of the nodes changes or a single node joins or leaves the system, the process has to be repeated from scratch. This means that none of this algorithm can be employed in data grid scales.

D. Beaver, S. Mycale, and P. Roadway has presented a protocol for secure mining of association rules in horizontally distributed databases. The protocol is designed based on Fast Distributed Mining (FDM) Algorithm and Secure Multiparty Algorithm. The Protocol offers enhanced privacy with respect to the protocol in [4]. In addition to that, it is simpler and is significantly more efficient in terms of communication rounds, computational cost and communication cost. Among the DM algorithms of interest to us is the discovery of association rules. The problem of discovering association rules was introduced by R. Agawam et.al. [1]

In the current scenario there has been growing attention in the area of distributed environment especially in data mining. Frequent pattern mining is active area of research in today’s scenario. PoonamModgiR. Srikant and R. Agrawal. Considered the problem of applying the Apriori algorithm to a more general class of attributes which could be either quantitative (e.g. age, income) or categorical (e.g. zip code, make of car)when dealing with distributed data sets, one important issue is the heterogeneity of the data. There has been considerable work in the database field dealing with heterogeneous databases and while we recognize this as an important topic, we are not looking into it directly. D. Agrawal and A. El Abaddi.

A.Ben-David, N. Nisan and B. Pinkas, have proposed a system for Multi Party Computation. Secure computation is one of the big achievements of modern cryptography, enabling a set of untrusted parties to compute any function of their private inputs while disclosing nothing but the result of the function. They have presented FairplayMP, a generic system for Secure Multiparty Computation. This is an extension of the Fair play system which supported secure computation by two parties. The extension to the multi-party case is needed since cryptographic protocols for the multi-party scenario are completely different from protocols for the two-party case [3].

M. Kantarcioglu and C. Clifton, have presented a protocol for Privacy-Preserving distributed mining of Association Rules on Horizontally Partioned data. The paper addresses the problem of computing association rules where the data may be distributed among various custodians, none of which are allowed to transfer their data to another site. Databases are homogeneous where all sites have the same schema but each site has information on different entities. Association Rules have been computed based on Fast Distributed Algorithm (FDM) and Secure Multiparty computation [1].JaideepVaidya, Chris Clifton, have proposed a protocol for privacy preserving Association Rule Mining in Vertically Partitioned Data.[21] The protocol is implemented through a two party algorithm for efficiently discovering frequent item sets with minimum support levels, without either site communicating individual transaction values. They have demonstrated that it is possible to achieve good individual security with communication cost comparable to that required to build a centralized data warehouse.[13]

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], [9]. 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 usual approach here is cryptographic rather than probabilistic. Liddell and Pinks [22] showed how-to securely build an ID3 decision tree when the training sets distributed horizontally. Lin et al. [16] discussed secure clustering using the EM algorithm over horizontally distributed data. The problem of distributed association rule mining was studied in [14], [21], [22] in the vertical setting, where each party holds a different set of attributes, and in [13]in the horizontal setting. Also the work of [17] 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 are computers that assist the resources to decrypt messages; another assumption made in [17] that distinguishes it from[13] and the present study is that no collusions occur between the different network nodes — resources or managers.

The problem of secure multiparty computation of the union of private sets was studied in [6], [11], [15], as well assign [13]. Freedman et al. [11] present a privacy-preserving protocol for set intersections. It may be used to compute also set unions through set complements, since A ∪ B = A ∩ B.Kissner and Song [15] present a method for representing sets as polynomials, and give several privacy-preserving protocols for set operations using these representations. They consider the threshold set union problem, which is closely related tithe threshold function (Definition 2.1). The communication overhead of the solutions in those two works, as well as in [13]'s and in our solutions, depends linearly on the size of the ground set. However, as the protocols in [11], [15] use homomorphic encryption, while that of [13] uses commutative encryption, their computational costs are significantly higher than ours. The work of Brick ell and Shmatikov [6] is an exception, as their solution entails a communication overhead that is logarithmic in the size of the ground set. However, they considered only the case of two players, and the logarithmic communication overhead occurs only when the size of the intersection of the two sets is bounded by a constant.

The problem of set inclusion can be seen as a simplified version of the privacy-preserving keyword search. In that problem, the server holds a set of pairs {(xi, pi)}in=1, wherein are distinct "keywords", and the client holds a single value. If w is one of the server's keywords, i.e., w = xi foursome 1 ≤ i ≤ n, the client should get the corresponding pain case w differs from all xi, the client should get notified of that. The privacy requirements are that the server gets no information about w and that the client gets no information about other pairs in the server's database. This problem was solved by Freedman ET. al. [10]. If we take all pi to be the empty string, then the only information the client gets is whether or not w is in the set {x1, . . . , xn}. Hence, in that case the privacy-preserving keyword search problem reduces to these inclusion problem. Another solution for the set inclusion problem was recently proposed in [20], using a protocol for oblivious polynomial evaluation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

III.METHODOLOGIES

A.Definitions and notations

Let D be a transaction database. As in [13], we view D as binary matrix of N rows and L columns, where each row Is a transaction over some set of items, $A = \{a1, . . . ,aL\}$, and each column represents one of the items in A. (In other words, the (i, j)th entry of D equals 1 if the ith transaction includes the item aj, and 0 otherwise.) The database D is partitioned horizontally between M players, denoted $P1, . . . , PM$. Player Pm holds the partial database Dmthat contains $Nm = |Dm|$ of the transactions in D, $1 \leq m \leq M$. The unified database is $D = D1 \cup \cdots \cup DM$, and it includes $N: =\Sigma M\ m=1\ Nm$ transactions. An itemset X is a subset of A. Its global support, $supp(X)$,is the number of transactions in D that contain it. Its local support, $suppm(X)$, is the number of transactions in Dmthat contain it. Clearly, $supp(X) =\Sigma Mm=1\ suppm(X)$. Let s be a real number between 0 and 1 that stands for a required support threshold. An itemset X is called s-frequent if $supp(X) \geq s\ N$. It is called locally s-frequent at $D\ m$ if $suppm(X) \geq sNm$.For each $1 \leq k \leq L$, let $Fk\ s$ denote the set of all k-itemsets (namely, itemsets of size k) that are s-frequent, and $Fk,m\ s$ be the set of all k-itemsets that are locally s-frequent at Dm,$1 \leq m \leq M$. Our main computational goal is to find, for a given threshold support $0 < s \leq 1$, the set of all s-frequent itemsets, $Fs:= \cup Lk=1\ Fk\ s$. We may then continue to find all (s, c)-association rules, i.e., all association rules of support at least $s\ N$ and confidence at least c. (Recall that if X and Y are two disjoint subsets of A, the support of the corresponding association rule $X \Rightarrow Y$ is $supp(X \cup Y)$ and its confidence is $supp(X \cup Y)/supp(X)$.)

B.The Fast Distributed Mining algorithm

The protocol of [13], as well as ours, are based on the Fast Distributed Mining (FDM) algorithm of Cheung et al.[8], which is an unsecured distributed version of the Apriori algorithm. Its main idea is that any s-frequent itemsetmustbe 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.

C.The FDM algorithm proceeds as follows:

(1) **Initialization**: It is assumed that the players have already jointly calculated $Fk-1s$. The goal is to proceed and calculate Fks.

(2) **Candidate Sets Generation:** Each player Pm computes the set of all $(k - 1)$-itemsets that are locally frequent in his site and also globally frequent; namely, Pm computes the set $Fk-1,ms\cap Fk-1s$. He then applies on that set the Apriori algorithm in order to generate the set Bk,ms of candidate k-itemsets.

(3) **Local Pruning:** For each $X \in Bk,ms$, Pm computes $suppm(X)$. He then retains only those itemsets that are locally$s$-frequent. We denote this collection of itemsets by Ck,ms.

(4) **Unifying the candidate itemsets:** Each player broadcasts his Ck,ms and then all players compute $Ck \cup s := Mm=1\ Ck,ms$.

(5) **Computing local supports.** All players compute the local supports of all itemsets in Cks.

(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 Cks.Finally,Fks is the subset of Cks that consists of all globally s frequent k-itemsets.

In the first iteration, when $k = 1$, the set $C1,ms$that the mth player computes (Steps 2-3) is just $F1,ms$, namely, the set of single items that are s-frequent in Dm. 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.

A running example

Let D be a database of $N = 13$ itemsets over a set of $L = 5$items, $A = \{1, 2, 3, 4, 5\}$. It is partitioned between $M = 3$ players, and the corresponding partial databases are:

$D1 = \{12, 12235, 124, 1245, 11, 112, 235, 24, 24\}$

$D2 = \{1223, 104, 23, 223, 2235\}$

$D3 = \{1223, 124, 104, 23\}$.

For example, $D1$ includes $N1 = 9$ transactions, the third of which (in lexicographic order) consists of 3 items — 1, 2 and4.Setting $s = 1/3$, an itemset is s-frequent in D if it is supported by at least $6 = sN$ of its transactions. In this case,
$F1$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

s = {1, 2, 3, 4}
*F*2
s = {12, 11, 23, 24, 23}
*F*3
s = {124}
*F*4
s = *F*5
s = ∅,
and *Fs*= *F*1*sF*2*s* ∪*F*3*s* .
For example, the itemset 23 is indeed globally *s*-frequent since it is contained in 6 transactions of
D. However, it is locally *s*-frequent only in *D*2 and *D*3.
In the first round of the FDM algorithm, the three players compute the sets *C*1,*ms*of all 1-itemsets that are locally frequent at their partial databases:
*C*1,1
s = {1, 2, 4, 5}, *C*1, 2
s = {1, 2, 3, 4}, *C*1, 3
s = {1, 2, 3, 4}.
Hence, *C*1
s = {1, 2, 3, 4, 5}. Consequently, all 1-itemsetshave to be checked for being globally frequent; that check reveals that the subset of globally *s*-frequent 1-itemsets is
*F*1
s = {1, 2, 3, 4}.
In the second round, the candidate item sets are:
*C*2,1
s = {12, 11, 24}
*C*2,2
s = {10, 11, 23, 24, 23}
*C*2,3
s = {12, 10, 11, 23, 24, 23} .
(Note that 12, 25, 45 are locally *s*-frequent at *D*1 but they are not included in *C*2,1*s*since 5 was already found to be globally infrequent.) Hence, *C*2*s* = {12, 10, 11, 23, 24, 23}.

Then, after verifying global frequency, we are left with *F*2
s ={12, 11, 23, 24, 23}.
In the third round, the candidate item sets are:
*C*3,1
s = {124} , *C*3,2
s = {223} , *C*3,3
s = {124} .
So, *C*3
s = {124, 223} and, then, *F*3
s = {124}. There are no more frequent item sets.

D.Secure Computation Of All Locally frequent Item sets

Here we discuss the secure implementation of Step 4 inthe FDM algorithm, namely, the secure computation of the union Cks =∪Mm=1 Skims. We describe the protocol of [13] and then our protocol. We analyze the privacy of the two protocols, and their computational cost

E.The protocol of Kantarcioglu and Clifton for the secure computation of all locally frequent item sets

Kantarcioglu and Clifton studied that problems and devised a protocol for its solution. 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

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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

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. Another problem of secure multiparty computation that we solve here as part of our discussion is 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. The main ingredient 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 holds.

1. Privacy Preserving Data Mining
2. Distributed Computation
3. Frequent Item sets
4. Association Rules

F.Privacy Preserving Data Mining

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. 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.

G.Distributed Computation

We compared the performance of two secure implementations of the FDM algorithm Section In the first implementation (denoted FDM-KC), we executed the unification step using Protocol UNIFI-KC, where the commutative cipher was 1024-bit RSA in the second implementation (denoted FDM) we used our Protocol UNIFI, where the keyed-hash function was HMAC.In both implementations, we implemented Step 5 of the FDM algorithm in the secure manner that was described in later.

We tested the two implementations with respect to three measures:

1) Total computation time of the complete protocols (FDMKC and FDM) over all players. That measure includes the Apriori computation time, and the time to identify the globally s-frequent item sets, as described in later.
2) Total computation time of the unification protocols only (UNIFI-KC and UNIFI) over all players.
3) Total message size. We ran three experiment sets, where each set tested the dependence of the above measures on a different parameter: • N — the number of transactions in the unified database.

H.Frequent Itemsets

We describe here the solution that was proposed by Kantarcioglu and Clifton. They onsidered 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 Δ(x) can be revealed for all. 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.

Association Rules

Once the set Fs of all s-frequent itemsets is found, we may proceed to look for all (s, c)-association rules (rules with support at least sN and confidence at least c). In order to derive from Fs all (s, c)-association rules in an efficient manner we rely upon the straightforward lemma

Overview

Protocol 1 is the protocol that was suggested by Kantarcioglu and Clifton [13] for computing the unified list of all locally frequent itemsets, *Ck s* =∪*Mm*=1 *Ck,m s* , without disclosing the sizes of the subsets *Ck,ms*nor their contents. The protocol is applied when the players already know *Fk*−1 *s* — the set of all(*k*−1)-itemsets that are globally *s*-

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

frequent, and they wish to proceed and compute *Fks*. We refer to it hereinafter as Protocol UNIFI-KC (Unifying lists of locally Frequent Itemsets — Kantarcioglu and Clifton).

The input that each player *Pm* has at the beginning of Protocol UNIFI-KC is the collection *Ck,m s* , as defined in Steps 2-3 of the FDM algorithm. Let *Ap*(*Fk*−1 *s*) denote the set of all candidate *k*-itemsets that the Apriori algorithm generates from *Fk*−1 *s* . Then, as implied by the definition of *Ck,m s* (see Section 1.1.2), *Ck,m s* , $1 \leq m \leq M$, are all subsets of *Ap*(*Fk*−1 *s*). The output of the protocol is the union *Ck s* = ∪*M m*=1 *Ck,m s* . In the first iteration of this computation *k* = 1, and the players compute all *s*-frequent 1-itemsets (here *F*0 *s*= *{}*). 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 *Ck s* the subset *Fk s* 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 *Fk s* is empty, the players may recover the full set *Fs*:= ∪*L k*=1 *Fk*of all globally *s*-frequent itemsets.

Protocol UNIFI-KC works as follows: First, each player adds to his private subset *Ck,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 1. An encryption algorithm is called commutative if *EK*1◦*EK*2 = *EK*2 ◦ *EK*1 for any pair of keys *K*1 and *K*2. 4 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. (Notation agreement: Since all protocols that we present herein involve cyclic communication rounds, the index *M* +1 always means 1, while the index 0always means M.)

IV.PROBLEM DEFINITION

What is a Secure Distributed Protocol?

The word “secure” under the context of this paper is related to the security definition of SMC.We also assume that parties are semi-honest in that they follow the execution requirement of the protocol but may use what they see during the execution to compute more than they need to know.(While the semi-honest model is a limitation, there are situations where a party will want to avoid knowing private data, to avoid the liability of protecting it – for such situations a semi-honest approach is appropriate.) Details of the security definitions and underlying models can be found in [12].

Definition 1 Let Ti be the input of party i,Q I (f) be the party i’s execution image of the protocol f and s be the result computed from f. f is secure if Qi(f) can be simulated from < Ti, s > and distribution of the simulated image is computationally indistinguishable from Qi(f).While it has been shown that for any polynomial time algorithm, there exists a polynomial time secure protocol that achieves the same functionality, generic solutions presented in [9, 36] require representing a problem as a Boolean circuit. On a large dataset, it is computationally impractical to adopt the pure circuit-based generic solution. Therefore, the key challenge in designing a secure and efficient protocol is to avoid using large circuits, instead the generic approach must be used only for small circuits (i.e., simple functionalities with compact inputs).

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 [22]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], [5], [12].In our problem, the inputs are the partial databases, and the required output is the list of association rules that hold inthe unified database with support and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

confidence no smaller T. Tassa is with the Department of Mathematics and Computer Science,The Open University, Ra'anana, Israel.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 onlyto 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. In such cases,some relaxations of the notion of perfect security might be inevitable when looking for practical protocols, provided that the excess information is deemed benign (see examples of such protocols in e.g. [13], [18], [19], [21], [22]).

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 usedin other contexts as well. Another problem of secure multiparty computation that we solve here as part of our discussionis the set inclusion problem; namely, the problem where Aliceholds 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.

At present, the most important association rule mining algorithm is Apriori put forward by R.Agrawal. It is an algorithm for mining the frequent itemsets. The Apriorialgorithm [1] is described as a fast algorithm for mining Association rules. The algorithm has the common structure of a two-step process. First all large itemsets that have support greater than minimum support are created incrementally. L1, the large itemsets of size 1 is created in first half over the data, by simply counting the appearance of each item in the data, Subsequent L's are created using their candidates. The candidates are potential large itemset of current size. The candidates are generated from the previous by using a fast and clever cross–product function. Then a pass over the data determines the candidate's support. All candidates with support>minsup are placed in the next L. This process stops when Ln is empty. These large itemsets are then used to produce rules. Each large itemset is divided into all Head =>Body combinations and each combination is tested for sufficient confidence. Here confidence is sup (itemset)/ sup (head). If the confidence tested is greater than the user defined minsup then the rule Head => Body is valid and is kept in a list.

V.EXPERIMENTAL ANALYSIS

We compared the performance of two secure implementations of the FDM algorithm (Section 1.1.2). In the first implementation(denoted FDM-KC), we executed the unification step (Step 4 in FDM) using Protocol UNIFI-KC, where the commutative cipher was 1024-bit RSA [25]; in the second implementation (denoted FDM) we used our Protocol UNIFI, where the keyed-hash function was HMAC [4]. In both implementations, we implemented Step 5 of the FDM algorithm inthe secure manner that was described in Section 3. We tested the two implementations with respect to three measures:

1) Total computation time of the complete protocols (FDMKCand FDM) over all players. That measure includes the Apriority computation time, and the time to identify the globally s-frequent item sets, as described in Section3. (The latter two procedures are implemented in the someway in both Protocols FDM-KC and FDM.)

2) Total computation time of the unification protocols only (UNIFI-KC and UNIFI) over all players.

3) Total message size.

We ran three experiment sets, where each set tested the dependence of the above measures on a different parameter

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

In our basic configuration, we took N = 500, 000, M = 10,and s = 0.1. 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. The results in each of those experiment sets are shown in Section 6.4.All experiments were implemented in mat lab and were executed on an Intel(R) Core(TM)i6-1715M personal computer with a 2.6GHz CPU, 8 GB of RAM, and the 64-bitoperating system Windows 6 Professional SP1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

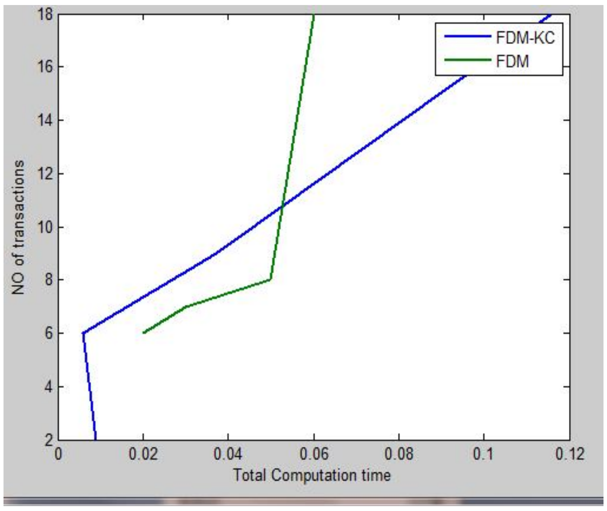

Figure 1

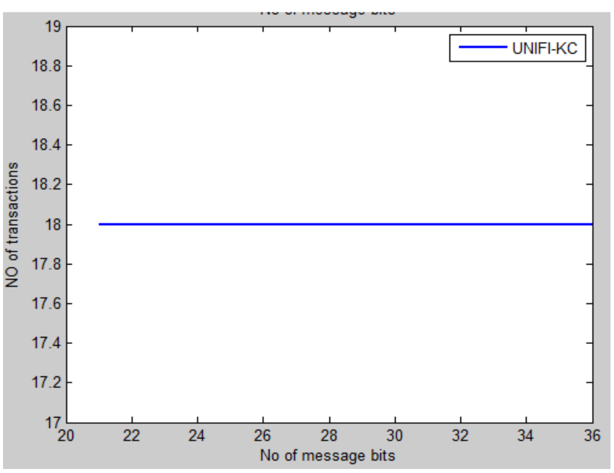

Figure 2

VI.CONCLUSION AND FUTURE WORK

We proposed a protocol for secure mining of association rules in vertically distributed databases that improves significantly upon the current leading protocol [13] 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. 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.

REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 486–499, 1494.
[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD Conference*, pages 439–450, 1500.
[3] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In *STOC*, pages 503–510, 1490.
[4] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Crypto*, pages 1–12, 1496.
[5] Ben-David, N. Nisan, and B. Pinkas. FairplayMP - A system for secure multi-party computation. In *CCS*, pages 256–176, 1508.

[6] J. Brickell and V. Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In *ASIACRYPT*, pages 236–252, 1505.
[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. In *PDIS*, pages 21– 42,

1496.
[8] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 21:469–462, 1485.
[9] A.V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *KDD*, pages 166–218, 1502.
[10] M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. In *TCC*, pages 203–324, 1505.
[11] M.J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, pages 1–14, 1504. .
[12] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16:1017–1036, 1504.
[13] M. Kantarcioglu, R. Nix, and J. Vaidya. An efficient approximate protocol for privacy-preserving association rule mining. In *PAKDD*, pages 512–524, 1509.
[14] L. Kissner and D.X. Song. Privacy-preserving set operations. In *CRYPTO*, pages 241–256, 1505.
[15] X. Lin, C. Clifton, and M.Y. Zhu. Privacy-preserving clustering with distributed EM mixture modeling. *Knowl. Inf. Syst.*, 8:68–81, 1505.
[16] Schuster, R. Wolff, and B. Gilburd. Privacy-preserving association rule mining in large-scale distributed systems. In *CCGRID*, pages 49– 413, 1504.
[17]T. Tassa and D. Cohen. Anonymization of centralized and distributed social networks by sequential clustering. *IEEE Transactions on Knowledge and Data Engineering*, 1512.
[18]T. Tassa and E. Gudes. Secure distributed computation of anonymized views of shared databases.*Transactions on Database Systems*, 36, Article 9, 1512.
[19]T. Tassa, A. Jarrous, and J. Ben-Ya'akov. Oblivious evaluation of multivariate polynomials. *Submitted*.
[20]J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *KDD*, pages 639–644, 1502.
[21]J. Zhan, S. Matwin, and L. Chang. Privacy preserving collaborative association rule mining. In *Data and Applications Security*, pages 123– 165, 1505.
[22]S. Zhong, Z. Yang, and R.N. Wright. Privacy-enhancing *k*-anonymization of customer data. In *PODS*, pages 109–116, 1505.