OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Securing Personal Information in Data Mining

[1]B. Sreenivasulu, [2]A. Sarat Kumar, [3]D. Srujan Chandra Reddy
[1] *Department of CSE, PBRVITS Kavali*
[2] *Department of CSE, PBRVITS Kavali*
[3]*Assoc. Prof Department of CSE, PBRVITS Kavali*

This paper presents few of the frequent authentication and security mechanisms and mechanisms used in several securing personal information in data mining techniques. The paper describes an overview of some of the well known securing personal information (privacy preserve) in data mining , - ID3 for decision tree, association rule mining, EM clustering, frequency mining and Naïve Bayes. Most of these algorithms are usually a modification of a well known data mining algorithm along with some privacy preserving techniques. The paper finally describes the problem of using a model without knowing the model rules on context of passenger classification at the airlines security checkpoint by homeland security. This paper is intended to be a summary and a high level overview of PPDM.

I. INTRODUCTION

Data mining refers to the techniques of extracting rules and patterns from data. It is also commonly known as KDD (Knowledge Discovery from Data). Traditional data mining operates on the data warehouse model of gathering all data into a central site and then running an algorithm against that warehouse. This model works well when the entire data is owned by a single custodian who generates and uses a data mining model without disclosing the results to any third party. However, in a lot of real life application of data mining, privacy concerns may prevent this approach. The first problem might be the fact that certain attributes of the data (SSN for example), or a combination of attributes might leak personal identifiable information. The second problem might be that the data is horizontally split across multiple custodians none of which is allowed to transfer data to the other site. The data might be vertically partitioned in which case, different custodians own different attributes of the data and they have the same sharing restrictions. Finally, the use of the data mining model might have restrictions, - some rules might be restricted, and some rules might lead to individual profiling in ways which are prohibited by law.

Privacy preserving data mining (PPDM) has emerged to address this issue. Most of the techniques for PPDM uses modified version of standard data mining algorithms, where the modifications usually using well known cryptographic techniques ensure the required privacy for the application for which the technique was designed. In most cases, the constraints for PPDM are preserving accuracy of the data and the generated models and the performance of the mining process while maintaining the privacy constraints. The several approaches used by PPDM can be summarized as below:

1. The data is altered before delivering it to the data miner.
2. The data is distributed between two or more sites, which cooperate using a semi-honest protocol to learn global data mining results without revealing any information about the data at their individual sites.
3. While using a model to classify data, the classification results are only revealed to the designated party, who does not learn anything else other that the classification results, but can check for presence of certain rules without revealing the rules.

In this paper, a high level overview of some of the commonly used tools and algorithms for PPDM is presented.

II. SECURE MULTI PARTY COMMUNICATION

Almost all PPDM techniques rely on secure multi party communication protocol. Secure multi party communication is defined as a computation protocol at the end of which no party involved knows anything else except it's own inputs the results, i.e. the view of each party during the execution can be effectively simulated by the input and output of the party. In the late 1980s, work on secure multi party communication demonstrated that a wide class of functions can be computed securely under reasonable assumptions without involving a trusted third party. Secure multi party communication has generally concentrated on two models of security. The semi-honest model assumes that each party follows the rule of the protocol, but is free to later use what it sees during execution of the protocol. The malicious model assumes that parties can arbitrarily cheat and such cheating will not compromise either security or the results, i.e. the results from the malicious party will be correct or the malicious party will be detected. Most of the PPDM techniques assume an intermediate model, -

preserving privacy with non-colluding parties. A malicious party may corrupt the results, but will not be able to learn the private data of other parties without colluding with another party. This is a reasonable assumption in most cases.

In the next section I'll present few efficient techniques for privacy preserving computations that can be used to support PPDM.

2.1 Secure Sum

Distributed data mining algorithms often calculate the sum of values from individual sites. Assuming three or more parties and no collusion, the following method securely computes such a sum.

Let $v = \sum_{i=1}^{s} v_i$ is to be computed for s sites and v is known to lie in the range [0..N]. Site 1, designated as the master site generates a random number R and sends $(R+v_1) \bmod N$ to site 2. For every other site l = 2, 3, 4 … s, the site receives:

$V = (R + \sum_{j=1}^{l-1} v_j) \bmod N$.

Site l computes:

$$(V + v_l) \bmod N = (R + \sum_{j=1}^{l} v_j) \bmod N$$

This is passed to site (l+1). At the end, site 1 gets:

$$V = (R + \sum_{j=1}^{s} v_j) \bmod N$$

And knowing R, it can compute the sum v. The method faces an obvious problem if sites collude. Sites (l-1) and (l+1) can compare their inputs and outputs to determine v_l . The method can be extended to work for an honest majority. Each site divides v_l into shares. The sum of each share is computed individually. The path used is permuted for each share such that no site has the same neighbors twice.

2.2 Secure Set Union

Secure set union methods are useful in data mining where each party needs to give rules, frequent itemsets, etc without revealing the owner. This can be implemented efficiently using a commutative encryption technique. An encryption algorithm is commutative if given encryption keys $K_1, K_2, ...K_n \in K$, the final encryption of a data M by applying all the keys is the same for any permuted order of the keys. The main idea is that every site encrypts its set and adds it to a global set. Then every site encrypts the items it hasn't encrypted before. At the end of the iteration, the global set will contain items encrypted by every site. Since encryption technique chosen is commutative, the duplicates will encrypt to the same value and can be eliminated from the global set. Finally every site decrypts every item in the global set to get the final union of the individual sets. One addition is to permute the order of the items in the global set to prevent sites from tracking the source of an item. The only additional information each site learns in the case is the number of duplicates for each item, but they cannot find out what the item is.

2.3 Secure Size of Set Intersection

In this case, every party has their own set of items from a common domain. The problem is to securely compute the cardinality/size of the intersection of these sets. The solution to this is the same technique as the secure union using a commutative encryption algorithm. All k parties locally generate their public key-part for a commutative encryption scheme. The decryption key is never used in this protocol. Each party encrypts its items with its key and passes it along to the other parties. On receiving a set of encrypted items, a party encrypts each item and permutes the order before sending it to the next party. This is repeated until every item has been encrypted by every party. Since encryption is commutative, the resulting values from two different sets will be equal if and only if the original values were the same. At the end, we can count the number of values that are present in all of the encrypted item sets. This can be done by any party. None of the parties can find out which of the items are present in the intersection set because of the encryption.

2.4 Scalar Product

Scalar product is a powerful component technique and many data mining problems can be reduced to computing the scalar product of two vectors. Assume two parties P_1 and P_2 each have a vector of cardinality n, $X = (x_1, x_2, ...x_n)$, $Y = (y_1, y_2, ...y_n)$. The problem is to securely compute $\sum_{i=1}^{n} x_i y_i$. There has been a lot of research and proposed solution to the 2 party cases, but these cannot be easily extended to the multi party case. The key approach to a possible solution proposed in [3] is to use linear combinations of random numbers to disguise vector elements and then do some computations to remove the effect of these random numbers from the result. Though this method does reveal more information than just the input and the result, it is efficient and suited for large data sizes, thus being useful for data mining.

2.5 Oblivious Transfer

The oblivious transfer protocol is a useful cryptographic tool involving two parties, - the sender and the receiver. The sender's input is a pair (x_0, x_1) and the receiver's input is a bit $\sigma \in (0,1)$. The protocol is such that the receiver learns x_σ (and nothing else) and the sender learns nothing. In the semi-honest adversaries, there exist simple and efficient protocols for oblivious transfer.

2.6 Oblivious polynomial evaluation

This is another useful cryptographic tool involving two parties. The sender's input is a polynomial Q of degree k over some finite field F (k is public). The receiver's input is an element $z \in F$. The protocol is such that the receiver learns Q (z) without learning anything else about the polynomial and the sender learns nothing.

In the next section, some common PPDM techniques are described:

III. ANONYMIZING DATA SETS

In many data mining scenarios, access to large amounts of personal data is essential for inferences to be drawn. One approach for preserving privacy in this case it to suppress some of the sensitive data values, as suggested in [5]. This is known as a k-anonymity model which was proposed by Samarati and Sweeney. Suppose we have a table with n tuples and m attributes. Let k > 1 is an integer. We wish to release a modified version of this table, where we can suppress the values of certain cells in the table. The objective is to minimize the number of cells suppressed while ensuring that for each tuple in the modified table there are k-1 other tuples in the modified table identical to it.

The problem of finding optimized k-anonymized table for any given table instance can be shown to be NP-hard even for binary attributes. There are however O(k) approximation algorithm discussed in [5] for solving this problem. The algorithm is also proven to terminate.

IV. DECISION TREE MINING

In the paper [4], a privacy preserving version of the popular ID3 decision tree algorithm is described. The scenario described is where two parties with database D_1 and D_2 wish to apply the decision tree algorithm on the joint database D_1 U D_2 without revealing any unnecessary information about their database. The technique described uses secure multi party computation under the semi honest adversary model and attempts to reduce the number of bits communicated between the two parties.

The traditional ID3 algorithm computes a decision tree by choosing at each tree level the best attribute to split on at that level ad thus partition the data. The tree building is complete when the data in uniquely partitioned into a single class value or there are no attributes to split on. The selection of best attribute uses information gain theory and selects the attribute that minimizes the entropy of the partitions and thus maximizes the information gain.

In the PPDM scenario, the information gain for every attribute has to be computed jointly over all the database instances without divulging individual site data. We can show that this problem reduces to privately computing x ln x in a protocol which receives x_1 and x_2 as input where $x_1 + x_2 = x$. This is described in [4].

V. ASSOCIATION RULE MINING

We describe the privacy preserving association rule mining technique for a horizontally partitioned data set across multiple sites. Let I = $\{i_1, i_2, ...i_n\}$ be a set of items and T = $\{T_1, T_2, ...T_n\}$ be a set of

transactions where each $T_i \subseteq I$. A transaction T_i contains an item set $X \subseteq I$ only if $X \subseteq T_i$. An association rule implication is of the form $X \Rightarrow Y$ ($X \cap Y = 0$) with support s and confidence c if s% of the transactions in T contains $X \cup Y$ and c% of transactions that contain X also contain Y. In a horizontally partitioned database, the transactions are distributed among n sites. The global support count of an item set is the sum of all local support counts. The global confidence of a rule can be expressed in terms of the global support:

$$SUP_g(X) = \sum_{i=1}^{n} SUP_i(X)$$

$$CONF_g(X \Rightarrow Y) = SUP_g(X \cup Y) \Big/ SUP_g(X)$$

The aim of the privacy preserving association rule mining is to find all rules with global support and global confidence higher than the user specified minimum support and confidence. The following steps, utilizing the secure sum and secure set union methods described earlier are used. The basis of the algorithm is the apriori algorithm which uses the (k-1) sized frequent item sets to generate the k sized frequent item sets. The problem of generating size 1 item sets can be easily done with secure computation on the multiple sites.

- Candidate Set Generation: Intersect the globally frequent item set of size (k-1) with locally frequent (k-1) itemset to get candidates. From these, use the Apiori algorithm to get the candidate k itemsets.
- Local Pruning: For each X in the local candidate set, scan the local database to compute the support of X. If X is locally frequent, it's included in the locally frequent itemset.
- Itemset Exchange: Compute a secure union of the large itemsets over all sites.
- Support Count: Compute a secure sum of the local supports to get the global support.

VI. EM CLUSTERING

Clustering is the technique of grouping data into groups called "clusters" based on the value of the attributes. A well known algorithm for clustering is the EM algorithm which works well for both discrete and continuous attributes. A privacy preserving version of the algorithm in the multi site case with horizontally partitioned data is described below.

Let us assume that the data is one dimensional (single attribute y) and are partitioned across s sites. Each site has n_l data items ($n = \sum_{l=1}^{s} n_l$). Let $z_{ij}^{(t)}$ denote the cluster membership for the ith cluster for the jth data point at the (t)th EM round. In the E step, the values μ_i (mean for cluster i), σ_i^2 (variance for cluster i) and π_i (Estimate of proportion of items i) are computed using the following sum:

$$\sum_{j=1}^{n} z_{ij}^{(t)} y_j = \sum_{l=1}^{s} \sum_{j=1}^{n_l} z_{ijl}^{(t)} y_j$$

$$\sum_{j=1}^{n} z_{ij}^{(t)} = \sum_{l=1}^{s} \sum_{j=1}^{n_l} z_{ijl}^{(t)}$$

$$\sum_{j=1}^{n} z_{ij}^{(t)} (y_j - \mu_i^{(t+1)})^2 = \sum_{l=1}^{s} \sum_{j=1}^{n_l} z_{ijl}^{(t)} (y_j - \mu_i^{(t+1)})^2$$ The second part of the summation in all these cases is local to every site. It's easy to see that sharing this value does not reveal y_i to the other sites. It's also not necessary to share n_l, and the inner summation values, but just computing n and the global summation for the values above using the secure sum technique described earlier.

In the M step, the z values can be partitioned and computed locally given the global μ_i, σ_i^2 and π_i. This also does not involve any data sharing across sites.

VII. FREQUENCY MINING

The basic frequency mining problem can be described as follows. There are n customers $U_1, U_2, ...U_n$ and each customer has a Boolean value d_i. The problem is to find out the total number of 1s and

0s without knowing the customer values i.e. computing the sum $\sum_{i=1}^{n} d_i$ without revealing each d_i. We cannot use the secure sum protocol because of the following restrictions.

- Each customer can send only one flow of communication to the miner and then there's no further interaction.
- The customers never communicate between themselves.

The technique presented in [8] uses the additively homomorphic property of a variant of the ElGamal encryption. This is described below:

Let G be a group in which discrete logarithm is hard and let g be a generator in G. Each customer U_i has two pairs of private/public key pair $(x_i, X_i = g^{x_i})$ and $(y_i, Y_i = g^{y_i})$. The sum $X = \sum_{i=1}^{n} X_i$ and $Y = \sum_{i=1}^{n} Y_i$, along with G and the generator g is known to everyone. Each customer sends to the miner the two values $m_i = g^{d_i}.X^{y_i}$ and $h_i = Y^{x_i}$. The miner computes $r = \prod_{i=1}^{n} \frac{m_i}{h_i}$. For the value of d for which $g^d = r$, we can show that this represents the sum $\sum_{i=1}^{n} d_i$. Since 0 < d < n, this is easy to find by encrypt and compare. We can also that assuming all the keys are distributed properly when the protocol starts, the protocol for mining frequency protects each honest customer's privacy against the miner and up to (n-2) corrupted customers.

VIII. NAÏVE BAYES CLASSIFIER

Naïve Bayes classifiers have been used in many practical applications. They greatly simplify the learning task by assuming that the attributes the independent given the class. They have been used successfully in text classification and medical diagnosis.

Naïve Bayes classification problem can be formulated as follows. Let $A_1, A, ...A_m$ be m attributes and V be the class attribute. Let each attribute A_i have a domain $\{a_i^1, a_i^2, ..., a_i^d\}$ and class attribute V has a domain $\{v^1, v^2, ..., v^d\}$. The data point for the classifier looks like $(a_{j1}, a_{j2}, ...a_{jm,} v_j)$. Given a new instance $(a_{j1}, a_{j2}, ...a_{jm,})$, the most likely class can be found using the equation:

$$v = \arg\max_{v^l \in V} P(v^l) \prod_{i=1}^{m} P(a_i \mid v^l)$$

This can be written is terms on number of occurrence # as:

$$v = \arg\max_{v^l \in V} \#(v^l) \prod_{i=1}^{m} \frac{\#(a_i, v^l)}{\#(v^l)}$$

The goal of the Privacy Preserving Naïve Bayes learner is to learn the Naïve Bayes classifier accurately, but the miner learns nothing about each customer's sensitive data except the knowledge derived from the classifier itself. To learn the classifier, all the miner needs to do is to learn $\#(v^l)$ and $\#(a_i, v^l)$ for each i, each k and each l. Since the occurrence of v^l or the pair (a_i, v^l) can be denoted by a Boolean value, we can use the technique described in Frequency Mining to compute the Naïve Bayes model with the privacy constraints.

IX. USING A MODEL WITHOUT DISCLOSING THE MODEL.

Recent homeland security measures uses data mining models to classify each airline passenger with a security tag. The problem statement comes from following requirements for the system:

- No one learns of the classification result other than the designated party.
- No information other than the classification result will be revealed to the designated party.
- Rules used for classification can be checked for certain condition without revealing the rules.

The problem can be formally stated as follows. Given an instance x from site D with v attributes, we want to classify x according to rule set R provided by site G. The rules $r \in R$ are of the form $\bigcap_{i=1}^{v}(L_i -> C)$, where each L_i is wither a clause $x_i = a$, or don't care (always true). Using the don't care clause, G can create arbitrary size rules and mask the actual number of clauses in the rule. In addition, D has a set of rules F that are not allows to be used for classification. The protocol will satisfy the following conditions:

- D will not be able to learn any rules in R
- D will be convinced that $F \bigcap R = 0$
- G will only learn the class value of x

The approach suggested in [2] uses a un-trusted non colluding site, where the only trust placed on the site is that it'll not collude with any of the other sites to violate privacy. Both G and D send synchronized streams of encrypted data and rule clause to site C. The orders of the attributes are scrambled in a way known to D and G only. Each attribute is given two values, - one corresponding to don't care and another it's true value. Each clause also has two values for every attribute. One is an "invalid" value to mask the real value and the other is the actual clause value or the "don't care" value. Site C compares both the values to see if the first or the second match. If yes, then either the attribute is a match or it's a "don't care". If there's a match for every clause in the rule, then the rule is true. To check for $F \bigcap R = 0$, commutative encryption technique is used and C compares the double encrypted version of the sets.

X. CONCLUSION

As usage of data mining for potential intrusive purposes using personally identifiable information increases, privately using these results will become more important. The above algorithm techniques show that it's possible to ensure privacy without compromising accuracy of results, and has a bound on the computation and the communication cost.

REFERENCES

[1]. Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data Murat Kantarcıoglu and Chris Clifton, Senior Member, IEEE
[2]. Assuring Privacy when Big Brother Murat Kantarcıoglu Chris Clifton
[3]. Privacy Preserving Association Rule Mining in Vertically Partitioned Data Jaideep Vaidya & Chris Clifton
[4]. Privacy Preserving Data Mining Yehuda Lindell & Benny Pinkasy
[5]. k-anonymity: Algorithm and Hardness, Gagan Aggarwal, Tomas Feder, Stanford University
[6]. Towards Standardization in Privacy Preserving Data Mining, Stanley R. M. Oliveira and Osmar R Zaiane, University of Alberta, Edmonton, Canada
[7]. Tools for Privacy Preserving Data Mining, Chris Clifton, Murat Kantarcioglu and Jaideep Vaidya, Purdue University.
[8]. Privacy Presercing Classification of Customer Data without Loss of Accuracy, Zhiqiang Yang, Sheng Zhong, Rebecca N. Wright.