Collaborative Research:
ITR: Distributed Data Mining to Protect Information Privacy

Chris Clifton (PI, Purdue University)
Wenliang (Kevin) Du (PI, Syracuse University)
Mikhail Atallah (Co-PI, Purdue University)

Data mining technology has emerged as a means of identifying patterns and trends from large quantities of data. Data mining has used a data warehousing model of gathering all data into a central site, then running an algorithm against that data. Privacy considerations may prevent this approach. For example, the Centers for Disease Control (CDC) may want to use data mining to identify trends and patterns in disease outbreaks, such as understanding and predicting the progression of a flu epidemic. Insurance companies have considerable data that would be useful – but are unwilling to disclose this due to patient privacy concerns. An alternative is to have each of the insurance companies provide statistics on their data that cannot be traced to individual patients, but can be used to identify the trends and patterns of interest to the CDC.

Privacy-preserving data mining has emerged to address this issue, with several papers in the past few years as well as articles in the popular press[23, 26]. One approach is to alter the data before delivering it to the data miner. The second approach assumes the data is distributed between two or more sites, and these sites cooperate to learn the global data mining results without revealing the data at their individual sites. This approach was first introduced to the data mining community by Lindell and Pinkas[32], with a method that enabled two parties to build a decision tree without either party learning *anything* about the other party's data, except what might be revealed through the final decision tree. We have since developed techniques for association rules[28, 44], decision trees with vertically partitioned data[21], clustering, k-nearest neighbor classification, and other methods are in progress.

In our previous research on this subject we have made two observations that should guide further work:

1. For each data mining approach, there are many *interesting* privacy-preserving distributed data mining problems. For example, Naïve Bayes is a classic approach to classification. However, the way the data is partitioned between parties, varied privacy constraints, and communication/computation considerations lead to *several* privacy-preserving solutions to Naïve Bayes.

2. While there may be many different data mining techniques, they often perform similar computations at various stages (e.g., counting the number of items in a subset of the data shows up in both association rule mining and learning decision trees.)

From observation 1, we feel the current approach of developing and publishing solutions to individual privacy-preserving data mining problems will generate more papers than real-world solutions. While such research is necessary to understand the problem, a myriad of solutions is difficult to transfer to industry. Observation 2 suggests an answer: build a *toolkit* of privacy-preserving distributed computation techniques, that can be *assembled* to solve specific real-world problems. If such component assembly can be simplified to the point where it qualifies as development rather than research, practical use of privacy-preserving distributed data mining will become widely feasible.

There are many variants of this problem, depending on how the data is distributed, what type of data mining we wish to do, and what restrictions are placed on sharing of information. Some problems are quite tractable, others are more difficult. Learning association rules with support and confidence thresholds provides an example. There is a simple distributed association rule mining algorithm that provides a degree of privacy to the individual sites. An example association rule could be:

> *Received Flu shot* and *age* > 50 implies *hospital admission*, where at least 5% of insured meet all the criteria (support), and at least 30% of those meeting the *flu shot* and *age* criteria actually

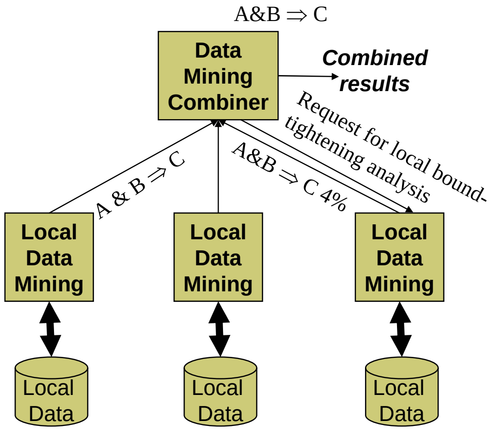

Figure 1: Example of computing association rules from individual site results.

require hospitalization (confidence).

There are algorithms to efficiently find all association rules with a minimum level of support. We can easily extend this to the distributed case using the following lemma: If a rule has *support* $> k\%$ globally, it must have *support* $> k\%$ on at least one of the individual sites.[1] A distributed algorithm based on this (from [11]) would be: Request that each site send all rules with support at least k. For each rule returned, request that all sites send the count of items they have that support the rule, and the total count of all items at the site. From this, we can compute the global support of each rule, and (from the lemma) be certain that all rules with support at least k have been found. An example of how this works is shown in Figure 1.

This algorithm does not require that the sites reveal individual data items. However, privacy constraints may be stricter than this. What if we want to protect not only the individual items at each site, but also how much each site supports a given rule? The above method reveals this information. Another variant where this approach fails is when the data is partitioned vertically: a single item may have part of its information at one site, and part at another.

1 Related Work

There are several research communities whose work can contribute to privacy preserving distributed data mining. We first discuss privacy preserving work in the data mining community, then related work from the cryptography and security communities, and finally distributed data mining work.

1.1 Privacy Preserving Data Mining

Privacy preserving data mining research has been divided into two types. The first is to alter the data before delivery to the data miner so that real values are obscured. If we add a random number chosen from a gaussian distribution to the real data value, the data miner no longer knows the exact value. However, important statistics on the collection (e.g., average) will be preserved. Research has addressed such statistical issues[33]. Recently data mining techniques on such altered data have been developed for constructing decision trees[3, 1] and association rules[37, 24]. This approach works in the "data warehouse" model of data mining, but trades off privacy for accuracy of results.

The second approach is privacy preserving distributed data mining. A secure multiparty computation based approach to building a decision tree was presented in [32]. Later work has addressed decision

[1] For proof of this, assume all the data is together, and divide data items into those that support the rule and those that do not. Next try to partition data such that no site has *support* $> k\%$. For each supporting item sent to a site, at least $1/k$ non-supporting items need to be sent to that site. We will run out of non-supporting items before assigning all the supporting items to sites.

trees in vertically partitioned data[21], and association rules in vertically partitioned[44] and horizontally partitioned[28] data. We propose to follow through on this approach, developing a toolkit of privacy preserving distributed computation methods and methods to compose them to produce privacy preserving distributed data mining techniques for various combinations of data mining task, data distribution, and privacy and security constraints.

1.2 Distributed Data Mining

There has been work in distributed data mining. Cheung et al. proposed a method for horizontally partitioned data[11], this is basically the approach outlined in the Figure 1. Other distributed data mining algorithms that partition the data into subsets have been developed[39, 41]. Recent work has addressed classification in vertically partitioned data [30], and situations where the distribution is itself interesting with respect to what is learned [45]. Work in parallel data mining may also be relevant [47, 29], some parallel mining algorithms also partition data[43]. Many of these techniques do not require that individual data items be exchanged between sites, however some do exchange samples of data[30].

Meta-learning for classification is another approach. With meta-learning, an ensemble of classifiers is used to get a global classifier. Distributed meta-learning techniques have been developed[8, 9, 35, 36]. Each site develops a classifier independently; these are used in concert to produce the "global classifier" results. This could protect the individual entities, but it remains to be shown that the individual classifiers do not release private information.

These algorithms often provide some type of privacy – individual data items may not be shared. However, *proof* that the techniques preserve privacy is lacking. Even if the methods avoid releasing individual data values, they may still reveal information about the collections at individual sites. In many applications this is unacceptable. Distributed data mining algorithms may provide a basis for privacy preserving distributed mining, as we shall show in Section 3.2.1, but methods are needed to adapt distributed algorithms to ones that provably preserve privacy.

1.3 Secure Multiparty Computation

The security and cryptography communities have set standards for what it means to *provably* maintain privacy and security. One concept that is particularly relevant to this proposal is *Secure Multiparty Computation*, introduced in [46]. The basic idea of Secure Multiparty Computation is that a computation is secure if at the end of the computation, no party knows anything except its own input and the results. One way to view this is to imagine a trusted third party – everyone gives their input to the trusted party, who performs the computation and sends the results to the participants. Now imagine we can achieve the same result without having a trusted party. Obviously, some communication between the parties is required for any *interesting* computation – how do we ensure that this communication doesn't disclose anything? The answer is to allow non-determinism in the exact values sent in the intermediate communication (e.g., encrypt with a randomly chosen key), and prove that a party using its own input and the result can generate a "predicted" intermediate computation that is as likely as the actual values.

There has been work in cooperative computation between entities that mutually distrust one another. This computation may be of any sort: scientific, data processing or even secret sharing. Secure two party computation was first investigated by Yao [46] and was later generalized to multiparty computation. The seminal paper by Goldreich proves that there exists a secure solution for *any* functionality[25]. The approach used is as follows: the function F to be computed is first represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Every participant gets random shares of the input and output wires for every gate. This approach, though appealing in its generality and simplicity, means that the number of rounds of the protocol grow with the size of the circuit. This grows with the size of the input. This is highly inefficient for large inputs, as in data mining. Although this proves secure solutions exist, achieving *efficient* secure solutions for distributed data mining is still open.

Secure Multiparty Computation makes two key contributions to the proposed work:

1. Methods for securely computing functions with small inputs (e.g., secure comparison), and

2. Definitions and proof techniques for private and secure computations in a distributed environment.

We will use item 2 as a basis for proving that the techniques we use will preserve privacy.

2 Research Problem

There are many scenarios where these issues arise. A few possible challenge problems are:

- Identifying public health problem outbreaks (e.g., epidemics, biological warfare instances). Many organizations collect data (insurance companies, HMOs, public health agencies). Individual privacy concerns will limit the willingness of the data custodians to share data, even with government agencies such as the Centers for Disease Control. These concerns extend beyond preserving patient privacy, as the data collectors (healthcare providers) may be unwilling to have anyone learn about the efficacy of their processes and procedures. Can we obtain the desired results while still preserving privacy of both individual entities and data collection sites?
- Collaborative corporations or entities. Ford and Firestone shared a problem with a jointly produced product: Ford Explorers with Firestone tires. Ford and Firestone may have been able to use association rule techniques to detect problems earlier. This would have required extensive data sharing. Factors such as trade secrets and agreements with other manufacturers stand in the way of needed sharing. Could we obtain the same results, while still preserving the secrecy of each side's data?

  Government entities face similar problems, such as limitations on sharing between law enforcement, intelligence agencies, and tax collection.
- Antitrust limitations on collaboration. The distinction between corporate collaboration and corporate collusion is hard to draw. Privacy preserving distributed data mining techniques would enable corporations to show that nothing is learned but the data mining results (presumably acceptable collaboration), thus eliminating concerns of possible illegal collusion hidden in the process.
- Multi-national corporations. An individual country's legal system may prevent sharing of customer data between a subsidiary and it's parent.

These examples each define a different problem, or set of problems. The problems can be characterized by the following three parameters:

Outcome. What is the desired data mining result? Do we want to *cluster* the data, as in the disease outbreak example? Are we looking for *association rules* identifying relationships among the attributes? There are several such data mining tasks, and each poses a new set of challenges. Sometimes it may be difficult (or impossible) to develop an efficient exact solution that meets the privacy constraints. The goal may be to obtain a solution with *bounded* error, or difference from the "data warehouse" solution.

Distribution. How is the data distributed? Is it *horizontally partitioned*, i.e., each entity is found only at a single site (as with medical insurance records)? Or do different sites contain different types of data (Ford on vehicles, Firestone on tires), giving *vertically partitioned* data?

Privacy. What are the privacy requirements? If the concern is solely that values associated with an individual entity not be released (e.g., "personally identifiable information") we will develop techniques that provably protect such information. In other cases, the notion of "sensitive" may not be known in advance. This would lead to human vetting of the intermediate results before sharing, requiring that individual site results be:

- compact;
- human-understandable; and
- complete (i.e., the method must not require numerous rounds of communication, each requiring human vetting.)

3 Research Plan

Each combination of Outcome, Distribution, and Privacy produces a different problem. Efficient solutions to each problem may require different algorithms, much as different threshold measures lead to different data mining algorithms (e.g., high support [2] versus high confidence [31] or high similarity [15]). Rather than developing a solution to each problem from scratch, we propose to develop privacy preserving components, a methodology for combining those components, and interact with researchers from other disciplines to demonstrate that this approach solves real-world problems. These will be concurrent efforts; real-world problems will be used to guide research into components and methods.

To demonstrate how this will work, we will first describe some privacy-preserving distributed computation components, as well as plans for future development in this area. We will then discuss plans to develop a methodology for component assembly. Finally, we will discuss ongoing and planned interactions with research areas where privacy preserving data mining will have direct impact.

3.1 Privacy Preserving Distributed Computation Components

Through our past research, we have identified several useful tools, such as data perturbation, 1-out-of-n oblivious transfer [19], homomorphic encryption [20], private permutation [20], scalar product [18, 19, 20], private comparison [5], and matrix operations [19]. However, those tools are not sufficient to address real-life problems. More tools need to be developed for preserving privacy during specific computations. We will investigate the abundant results in the areas of cryptography, secure multi-party computation, data perturbation, and data disguise.

We present several representative efficient methods for privacy-preserving computations that can be used to support data mining. Not all are truly secure multiparty computations – in some, information other than the results is revealed – but all do have provable bounds on the information released. In addition, they are *efficient*: the communication and computation cost is not significantly increased through addition of the privacy preserving component. However, since these building blocks will be used for many times in data mining computations, we also plan to develop more efficient ways to conduct these computations.

This is not an exhaustive list of efficient secure multiparty computations. Some other examples can be found in [5, 19]. The ones given below allow us to present several privacy-preserving solutions to data mining problems in Section 3.2. Additional problems to be addressed include top-K queries, k-median, comparison, multi-party scalar product, and computing Entropy and Gini Index [21]. As the project proceeds, we will identify and address new secure computations that enable privacy preserving data mining.

3.1.1 Secure Sum

Secure sum is often given as a simple example of secure multiparty computation[42]. We mention it here because of its applicability to data mining (see Sections 3.2.1 and 3.2.3). A simple solution to computing $sum_{i=1}^{n} v_i$ is for the first site to choose a random value R, then send $x_1 = v_1 + R$ to site 2. The remaining sites $2..n$ receive x_{i-1}, and send $x_i = x_{i-1} + v_i$ to the next site, with site n sending the result to site 1. Site 1 subtracts the R to get the result.

The random value R disguises all the intermediate sums, hiding the actual values. Even this simple algorithm is surprisingly subtle to *prove* secure, for more details see [13].

3.1.2 Scalar Product

Many data mining problems can essentially be reduced to computing the scalar product. One example of this, reducing association rule mining to scalar product computation, will be discussed in Section 3.2.2. We present several techniques for computing the scalar product of two vectors, showing a range of approaches to disguising the data. In what follows, let $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$; $\hat{X}$ represents the disguised X and $\hat{Y}$ represents the disguised Y.

Linear Transformation Disguise Scheme 1: The key insight of this approach (from [44]) is to use linear combinations of random numbers to disguise vector elements, compute the product of the disguised vectors, then remove the effect of these random numbers from the result.

Briefly, party A starts with X, party B with Y, and both agree on an $n \times n/2$ matrix A of coefficients $a_{i,j}$. A generates $n/2$ random numbers $R_1 \dots R_{n/2}$, and using this creates the following $\hat{X}$ to send to B:

$$\begin{gathered}\langle x_1 + a_{1,1} * R_1 + a_{1,2} * R_2 + \dots + a_{1,n/2} * R_{n/2} \rangle \\ \langle x_2 + a_{2,1} * R_1 + a_{2,2} * R_2 + \dots + a_{2,n/2} * R_{n/2} \rangle \\ \dots \\ \langle x_n + a_{n,1} * R_1 + a_{n,2} * R_2 + \dots + a_{n,n/2} * R_{n/2} \rangle\end{gathered}$$

B computes $\hat{S} = \hat{X} \cdot Y$, and sends $\hat{S}$ and the following $\hat{Y}$ to A:

$$\begin{gathered}\langle a_{1,1} * y_1 + a_{2,1} * y_2 + \dots + a_{n,1} * y_n \rangle \\ \langle a_{1,2} * y_1 + a_{2,2} * y_2 + \dots + a_{n,2} * y_n \rangle \\ \dots \\ \langle a_{1,n/2} * y_1 + a_{2,n/2} * y_2 + \dots + a_{n,n/2} * y_n \rangle\end{gathered}$$

$\hat{S}$ can be written as:

$$\begin{aligned}\hat{S} = \quad \sum_{i=1}^{n} x_i * y_i \quad & +R_1 * (a_{1,1} * y_1 + a_{2,1} * y_2 + \dots + a_{n,1} * y_n) \\ & +R_2 * (a_{1,2} * y_1 + a_{2,2} * y_2 + \dots + a_{n,2} * y_n) \\ & \dots \\ & +R_{n/2} * (a_{1,n/2} * y_1 + a_{2,n/2} * y_2 + \dots + a_{n,n/2} * y_n)\end{aligned}$$

(see [44] for details). $\sum_{i=1}^{n} x_i * y_i$ is the desired final result. The rest is $R \cdot \hat{Y}$, which A can compute and subtract from $\hat{S}$ to get the desired result. Since each side receive too few equations to solve for the unknown values, neither side can determine the other's x_i or y_i. Though this method reveals more than just the input and the result, it preserves privacy of X and Y, and has reasonable computation and communication costs.

Linear Transformation Disguise Scheme 2: Assume M is an invertible n by n matrix in real domain, then $X' = XM$ transforms X to another vector. If $m < n$ elements of X' is disclosed to someone, it is impossible to derive the raw data in X because the number of possibilities is infinite if X is in real domain. Using this fact, we can design the following solution to the scalar product problem: Let $X' = XM$, $Y' = M^{-1}Y$, and note that $X \cdot Y = X' \cdot Y'$. Let $X' = (X^-, X_-)$, where X^- represents the first half of the vector X' and X_- represents the other half of X'. Similarly, let $Y' = (Y^-, Y_-)$. To conduct the scalar product between X' and Y', Alice discloses X^- to Bob, while Bob discloses Y_- to Alice. Then Alice computes $X_- \cdot Y_-$, and Bob computes $X^- \cdot Y^-$; the sum of these two scalar products equals to $X \cdot Y$. Because of the property of the linear transformation, neither party can determine the raw data of the other party's vector.

Disguise Using A Commodity Server Sometimes, with a third party (still untrusted) that does not collude with any participant, the computation might become much more efficient. We describe a special third party, the *commodity server*, which has been used in secure multiparty computation work.

With the commodity server, participants can send request to the commodity server and receive data (called *commodities*). There are two requirements on the commodities. First the commodities should be independent of participant's private data; second the commodities should be independent of the data sent to the commodity server during the interaction. This means the commodity server can generate independent data off-line, and sell them as commodities to clients (hence the name "commodity server").

The Commodity-Based architecture was first proposed by Beaver [7, 6], and has been used for solving secure multiparty computation problems in the literature [7, 6, 16, 17, 21]. This architecture has appealing features: First, the server does not participate in the computation, but provides data enably the participants to hide their private data. Second, the data provided by the server does not depend on the clients' private data; therefore the commodity server need not know the private data. This reduces the liability risk of the third-party server if a client's private data is somehow disclosed. Furthermore, this feature make it easy to find such a server because not much trust is needed for a third-party server. Of course, the drawback of the commodity server is the assumption that there will be no collusion between the commodity server and any participant.

(Scalar Product Protocol–Commodity Server Approach)
Inputs: Alice has a vector $X = (x_1, \ldots, x_n)$, and Bob has a vector $Y = (y_1, \ldots, y_n)$.
Outputs: Alice gets the scalar product $X \cdot Y = \sum_{i=1}^{n} x_i y_i$.

1. The Commodity Server generates a pair of random vectors R_a and R_b, and let $r_a + r_b = R_a \cdot R_b$, where r_a (or r_b) is a random number. Then the server sends R_a and r_a to Alice, sends R_b and r_b to Bob.
2. Alice sends $\hat{X} = X + R_a$ to Bob, and Bob sends $\hat{Y} = Y + R_b$ to Alice.
3. Bob computes $\hat{X} \cdot Y + r_b$, then sends the result to Alice.
4. Alice computes $(\hat{X} \cdot Y + r_b) - (R_a \cdot \hat{Y}) + r_a = X \cdot Y + (r_b - R_a \cdot R_b + r_a) = X \cdot Y$.

In this research plan, we need to investigate whether these data disguising techniques can be applied to other computations used in various data mining. In addition, we will identify and develop more data disguise techniques as we gain more understanding of the data mining computations. One particular problem that will be addressed is the m-party case of scalar product. While this generalized problem is easily stated, none of the suggested solutions can be easily extended to solve the multi-party problem.

3.1.3 Secure Set Union

Secure union methods are useful in data mining where each party needs to give rules, frequent itemsets, etc., without revealing the owner. The union of items can be evaluated using general secure multiparty computation methods if the domain of the items is small. Each party creates a binary vector where 1 in the i^{th} entry represents that the party has the i^{th} item. After this point, a simple circuit that *or's* the corresponding vectors can be built and it can be securely evaluated using general secure multi-party circuit evaluation protocols. However, this requires communication costs on the order of the size of the domain - an unreasonable requirement for practical data mining. To overcome this problem a simple approach based on commutative encryption is used. An encryption algorithm is commutative if given encryption keys $K_1, \ldots, K_n \in K$, for any m in domain M, and for any permutation i, j, the following two equations hold:

$$E_{K_{i_1}}(\ldots E_{K_{i_n}}(M)\ldots) = E_{K_{j_1}}(\ldots E_{K_{j_n}}(M)\ldots) \tag{1}$$

$\forall M_1, M_2 \in M$ such that $M_1 \neq M_2$ and for given k, $\epsilon < \frac{1}{2^k}$

$$Pr(E_{K_{i_1}}(\ldots E_{K_{i_n}}(M_1)\ldots) = E_{K_{j_1}}(\ldots E_{K_{j_n}}(M_2)\ldots)) < \epsilon \tag{2}$$

With shared p the Pohlig-Hellman encryption scheme[34] satisfies the above equations, but any commutative encryption scheme can be used.

Each site encrypts its items, then encrypts the items from other sites until each site has encrypted every item. Since equation 1 holds, once each site has encrypted every item, duplicates in the original items will be duplicates in the encrypted items and can be deleted. Due to equation 2, only duplicates will be deleted. Every site then decrypts in turn, revealing the result. Decryption can occur in any order, so by permuting the encrypted items we prevent sites from tracking the source of an item. An example is shown in Figure 2.

3.1.4 Secure Size of Set Intersection

Consider several parties having their own sets of items from a common domain. The problem is to securely compute the cardinality/size of the intersection of these local sets. Formally, given k parties $P_1 \ldots P_k$ having local sets $S_1 \ldots S_k$, we wish to securely compute $|S_1 \cap \ldots \cap S_k|$. We can do this using a parametric commutative one way hash function. One way of getting such a hash function is to use commutative public key encryption, such as Pohlig-Hellman, and throw away the decryption keys. Commutative encryption has already been described in Section 3.1.3.

All k parties locally generate their public key-pair (E_i, D_i) for a commutative encryption scheme. (They can throw away their decryption keys since these will never be used.) 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

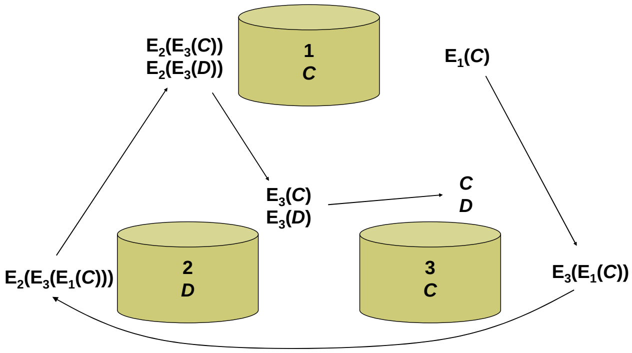

Figure 2: Determining the Union of a set of items.

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 (i.e., the item was present in both sets). Thus, we need only count the number of values that are present in *all* of the encrypted itemsets. This can be done by any party. Since items are never decrypted, no party knows which items are present in the intersection set.

This algorithm is not secure under Secure Multiparty Computation definitions. The number of items present at each site is revealed. However, *any* solution that would be a true secure multiparty computation would require that each site send enough (possibly encrypted) data to represent all possible items, making completely secure solutions impractical for data mining. This brings out an interesting tradeoff that will guide this research: obtaining efficient solutions with provable privacy properties.

3.2 Combination Methodology

We now demonstrate how the above protocols can be used to make several standard data mining algorithms into privacy-preserving distributed data mining algorithms. Others we plan to work on include linear programming under various combinations of data partitioning and privacy constraints clustering and multiparty classification in vertically partitioned data, time series mining, anomaly detection, and text mining. We will study several classification schemes including decision trees, Bayesian classifiers, genetic algorithm, neural networks, and Support Vector Machines. For example, we plan to investigate securely constructing an approximate Support Vector Machine by approximating the first few summation terms of the kernel integral, converting it into a scalar product problem (Section 3.1.2).

3.2.1 Association rules in horizontally partitioned data

We address association rule mining as defined in [2]: Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items and DB be a set of transactions, where each transaction $T \in DB$ 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 $X \Rightarrow Y$ where $X \subseteq I, Y \subseteq I$ and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ has *support* s in the transaction database DB if $s\%$ of the transactions in DB contain $X \cup Y$. The rule has *confidence* c if $c\%$ of the transactions in DB that contain X also contains Y. An itemset X with k items is called a k-itemset. The problem of mining association rules is to find all rules whose support and confidence are higher than a specified minimum support and confidence.

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 the local support counts. An itemset X is *globally supported* if the global support count of X is bigger than $s\%$ of the total transaction database size. The global confidence of a rule $X \Rightarrow Y$ can be given as $\{X \cup Y\}.sup/X.sup$. A k-itemset is called a globally large k-itemset if it is globally supported.

The FDM algorithm [10] is a fast method for distributed mining of association rules:

1. **Candidate Set Generation**: Intersect the globally large itemsets of size $k-1$ with locally large $k-1$ itemsets to get candidates. From these, use the classic apriori candidate generation algorithm to get

the candidate k itemsets.

2. **Local Pruning**: For each X in the local candidate set, scan the local database to compute the local support of X. If X is locally large, it is included in the locally large itemset list.
3. **Itemset Exchange**: Broadcast locally large itemsets to all sites – the *union* of locally large itemsets, a superset of the possible global frequent itemsets. (It is clear that if X is supported globally, it will be supported at least at one site.) Each site computes (using apriori) the support of items in union of the locally large itemsets.
4. **Support Count Exchange**: Broadcast the computed supports. From these, each site computes globally large k-itemsets.

The above algorithm avoids disclosing individual transactions, but does expose significant information about the rules supported at each site. Using the Secure Union of Section 3.1.3 for step 3, and secure sum for step 4, we can compute global support without revealing individual supports. The confidence of large itemsets can also be found using this method. We would like to emphasize that if the goal is to have a totally secure method, the union step would have to be eliminated. However, using the secure union method gives higher efficiency with provably controlled disclosure of some minor information (i.e., the number of duplicate items and the candidate sets.) Full discussion of this method can be found in [28].

3.2.2 Association rules in vertically partitioned data

Mining private association rules from vertically partitioned data, where the items are partitioned and each itemset is split between sites, can be done by extending the apriori algorithm. Most of the apriori algorithm can be done locally at each of the sites. The crucial step involves finding the support count of an itemset.

Consider the entire transaction database to be a boolean matrix where 1 represents the presence of that item (column) in that transaction (row), while 0 correspondingly represents an absence. The key insight is as follows: The support count of an itemset is *exactly* the scalar product of the vectors representing the sub-itemsets with both parties. Thus, if we can compute the scalar product securely, we can compute the support count. Full details are given in [44].

Another way of finding the support count is as follows: Let party i represent its sub-itemset as a set S_i which contains only those transactions which support the sub-itemset. The support count is the size of the intersection set of all these local sets $|\cap_{i=1}^{n} S_i|$, computed as in Section 3.1.4.

These protocols assume a semi-honest model, where the parties involved will honestly follow the protocol but can later try to infer additional information from whatever data they receive through the protocol. One result of this is that parties are not allowed to give spurious input to the protocol. If a party is allowed to give spurious input, they can probe to determine the value of a specific item at other parties. For example, if a party gives the input $(0, \ldots, 0, 1, 0, \ldots, 0)$, the result of the scalar product (1 or 0) reveals the value of the "private" data corresponding to the 1. Attacks of this type are termed probing attacks. All of the protocols currently suggested in the literature are susceptible to probing attacks. Better techniques which work even in the malicious model are needed to guard against this.

3.2.3 EM Clustering

We present a privacy preserving EM algorithm for secure clustering. Only the one dimensional case is shown; extension to multiple dimensions is straight forward. The convention for notations is given below:

k	Total number of mixture components (clusters).	s	Total number of distributed sites.
n	Total number of data points.	n_l	Total number of data points for site l.
y_j	Observed data points.	μ_i	Mean for cluster i.
σ_i^2	Variance for cluster i.	π_i	Estimate of proportion of items in cluster i.
z_{ij}	Cluster membership. If $y_j \in$ cluster i, $z_{ij} \approx 1$, else $z_{ij} \approx 0$.		

i, j, l are the indexes for the mixture component, data points and distributed sites respectively. t denotes the iteration step.

From conventional EM mixture models for clustering, we assume that data y_j are partitioned across s sites $(1 \leq l \leq s)$. Each site has n_l data items, where summation over all the sites gives n. To obtain a global estimation for $\mu_i^{(t+1)}$, $\sigma_i^{2(t+1)}$, and $\pi_i^{(t+1)}$ (the E step) requires only the global values n and

$$\begin{aligned}\sum_{j=1}^{n} z_{ij}^{(t)} y_j &= \sum_{l=1}^{s}\sum_{j=1}^{n_l} z_{ijl}^{(t)} y_j \\ \sum_{j=1}^{n} z_{ij}^{(t)} &= \sum_{l=1}^{s}\sum_{j=1}^{n_l} z_{ijl}^{(t)} \\ \sum_{j=1}^{n} z_{ij}^{(t)}(y_j - \mu_i^{(t+1)})^2 &= \sum_{l=1}^{s}\sum_{j=1}^{n_l} z_{ijl}^{(t)}(y_j - \mu_i^{(t+1)})^2\end{aligned}$$

Observe that the second summation in each of the above equations is local. Using secure sum (Section 3.1.1) we can compute the global values securely, without revealing y_j.

The estimation step giving $\mathbf{z}$ can be partitioned and computed locally given global μ_i, σ_i^2, and π_i:

$$z_{ijl}^{(t+1)} = \frac{\pi_i^{(t)} f_i(y_j; \mu_i^{(t)}, \sigma_i^{2(t)})}{\sum_i \pi_i^{(t)} f_i(y_j; \mu_i^{(t)}, \sigma_i^{2(t)})}$$

where y_j is a data point at site l. The E-step and M-step iterate until

$$|L^{(t+1)} - L^{(t)}| \leq \epsilon,$$

where

$$L^{(t)}(\theta^{(t)}, \mathbf{z}^{(t)}|y) = \sum_{j=1}^{n}\sum_{i=1}^{k}\{z_{ij}^{(t)}[\log \pi_i f_i(y_j^{(t)}|\theta^{(t)})]\}.$$

Again, this can be computed using a secure sum of locally computed partitions of $\mathbf{z}$.

3.2.4 K-Nearest Neighbor Classification

Classification introduces a privacy constraint that is not easily addressed in the Secure Multiparty Computation model. If the target class in the training data is private information, the classifier could be used to undermine privacy. One solution is not to build a global classifier. Instead, the parties collaborate to classify each instance. Therefore, the problem becomes how to share a classifier (a secret) among the various distributed sites while still supporting the classification function.

We are currently developing methods for secure distributed k-nn classification. The problem can be defined as follows: Given the publicly available instance to be classified, find the most common class label of the k nearest attributes without revealing which class label originated at which data site. To accomplish this efficiently, we propose to use an **untrusted**, **non-colluding** commodity server.

Currently we have two different solutions for different level of security needs. One is provably secure under certain cryptographic assumptions. The other is secure under the assumption that finding the roots of the large degree polynomials are computationally hard.

Both methods are using the secure union algorithm to form a set of $k * n$ points where each of the n sites contributes k points. A series of secure comparisons are done to get the k data points among $k * n$ points that are closest to given data instance. Size of secure set intersection can be used to find the majority class. Under reasonable assumptions, our methods guarantee that except the site that provides the data instance for classification, no site learns the final result.

3.3 Quantifying Privacy

One focus of this project will be to understand and define privacy and security in ways that make sense for data mining. The secure multiparty computation approach has two limitations:

1. It is too restrictive; truly secure solutions may be inefficient. (E.g., for set intersection to be completely secure, each site must send enough data to represent all possible values, even if much is just “dummy” data).

2. It doesn't guarantee privacy. It only guarantees that nothing is disclosed beyond the result, but what if the result itself violates privacy?

We need ways to define and measure privacy to ensure that privacy preserving data mining results do meet actual privacy constraints. Sketches of several approaches are given below (more details in [14]):

Bounded Knowledge. Approaches that alter the data generally use a bounded knowledge definition of privacy, perhaps the best method to date is the entropy based metric of [1]. While secure multiparty computation appears to achieve "perfect" privacy, in that nothing is shared but the results, even the results can provide bounded knowledge on the data sources.

Need to Know is well established in controlling access to data. In the U.S., access to classified data requires both a security clearance and a justification of why the data should be accessed. The same concept appears in the EC95/46 directive[22]:

> Member States shall provide that personal data may be processed only if:
> (a) the data subject has unambiguously given his consent; or
> (b) processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract; or ...

Note clause (b): It is acceptable to use individual data if it is needed to achieve a result requested by the individual.

A need to know standard can be used to balance the tradeoff between results and the potential compromise of privacy by those results.

Protected from disclosure. We may want to protect specific items: individual data items, or specific rules[4, 40]. The problem becomes more difficult when we want to protect against disclosure of classes of information – in the limit this prevents data mining altogether[12]. We will develop privacy measures to address this issue; a likely starting point is ability to learn a classifier for a protected attribute from the results.

Anonymity is an established measure of privacy, including concepts such as k-anonymity[38]. We have proposed a p-indistinguishability metric that extends this concept to data mining, allowing results that reveal information about an individual as long as the results reveal equivalent information for *all* individuals.

The proposed project will formalize these measures and use them to analyze the developed privacy preserving data mining constructs. We will also investigate the applicability of these measures, and identify and formalize new measures as appropriate.

3.4 Transfer to real-world problems

While we plan to use the traditional technology "push" mechanism of publishing articles in journals and conferences, we believe technology "pull" will speed the impact of this work. To this end, we are working with the Center for E-Business Education and Research at Purdue's Krannert School of Management with seed funding provided through the Lilly Foundation. Through this collaboration we have identified several privacy-preserving variants of linear programming that have widespread applicability to logistics problems. If we can efficiently approximate existing linear programming solutions in a privacy preserving manner, the techniques will have immediate impact on optimization in a variety of industrial and logistics domains.

We have begun discussions with the DuPont LYCRA division and Ford Motor Company on obtaining real-world data to use as research testbeds. Through discussions with the companies, we will identify problems that are significant to them. We will then develop experiments based on their own data that simulate their real problems. Although we do not expect to field these prototypes as part of this research effort, we do plan to use these corporate contacts to increase opportunities for student internships. If we are able to identify real-world problems where our prototypes can be easily applied, we will encourage the companies to use student internships as a way to apply the prototypes to their problems.

Another source of problems, and potential technology transfer path, is the Indiana Telemedicine Incubator [27]. This is a consortium for the development of distributed, multimedia database technology for the health care industry. Data mining can be used to significantly improve health care, however patient data needs to be kept private. This should be a fertile ground for identifying significant challenge problems. Since the project is managed by the Indiana Center for Database Systems at Purdue, strong potential exists to test distributed data mining prototypes on Indiana Telemedicine Incubator systems. Corporate members of the Telemedicine project (such as Micro Database Systems, Inc.) have already commercialized technology developed at the Indiana Center for Database Systems, and will provide a channel for transferring successful privacy-preserving data mining prototypes into commercial systems. We are also discussing this technology with the medical informatics experts at the Regenstrief Institute and Indiana University School of Medicine to determine applicability to the health care field.

3.5 Timeline

The first year of this effort will concentrate on solutions to problems already identified, such as linear programming. We will also work on formal proof of the efficacy and complexity of unpublished approaches outlined in Section 3.1, and prototyping the techniques of Section 3.2 to support validation on simulated datasets. The anonymity measure outlined in 3.3 will be formalized, and the approaches described in Section 3.2 will be evaluated against these measures.

Throughout the project we will identify new challenge problems, as described in Section 3.4. In the first year we will formalize privacy preserving data mining problems in anomaly detection. Data that is globally anomalous may be common at one site – this site would not know it has unusual data without distributed data mining. One approach is to have each site compute aggregates, and pass these to a central site that determines global aggregate values. Local sites could then find anomalies with respect to the global aggregates. Computer security provides real-world anomaly detection problems; we will work with Purdue's Center for Education and Research in Information Assurance and Security to identify such challenges.

During the second year of the project, the problems addressed in the first year will be expanded, to develop a family of problems and solutions. We will formalize other privacy measures from Section 3.3. The initially developed algorithms and prototypes will be tested against real-world data. We plan to request funding through the Post Doctoral Research Associates in Experimental Computer Science program under the Experimental and Integrative Activities division to support the prototyping and testing efforts. Results will be used to encourage other funding sources (particularly corporate) to support technology transfer efforts. New problem areas for the third-year effort will be identified.

The third year will focus on two issues: theoretic foundation and applications. For the theoretic foundation, we will develop a general framework to quantify privacy that goes beyond the measures of Section 3.3. We want to be able to evaluate a solution by measuring how much private information has been disclosed, as well as by measuring its computation and communication complexity. We will also look for real-life applications of our results. Examples of these applications include distributed intrusion detection, collaborative forensics, and collaborative credit card fraud detection. The plan is to develop a self-sustaining research program, where new problems will be continuously identified and solved on a three-year (dissertation) cycle:

1. Define challenge problem, develop preliminary solution, and identify test data sets.
2. Expand to a family of similar problems, refine solution, and prototype against simulated data.
3. Test against real data, formalize results, and develop technology transfer plan.

This will continue until the toolkit and methodology are complete and solid enough that reasonably knowledgeable professionals can use them to develop solutions to new problems.

4 Success Metrics

What measures of success are applicable to this work? In addition to standard data mining measures such as accuracy of the result and computation cost, we face issues related to distribution and security. Each of these communities has measures for the success of work in their area. A good solution to a privacy preserving distributed data mining problem must meet several criteria:

Quality of results. There are standard techniques in the machine learning community to evaluate the efficacy of an algorithm against a known dataset. While we will evaluate results in this manner, we believe a better approach for this work is to evaluate in terms of known *methods*. Our goal will be to develop algorithms that perform within a bounded error of established centralized data mining approaches. For example, we may be able to prove that a distributed privacy preserving clustering algorithm produces clusters where each cluster contains at least 95% of the items that would fall into the cluster produced by *k-means* clustering.

Computational cost. Standard computational measures, such as worst-case running time, are often inappropriate for data mining. Algorithms that work well in practice may have intractable worst-case running time or space requirements, but data causing worst-case performance may be rare or nonexistent in practice. Instead, we plan to measure costs in relation to existing data mining algorithms, e.g., "$O(\log k)*$a-priori running time, where k is chosen depending on the privacy required".

Communication cost. We see two primary measures for communication cost: Total bits, and number of messages. Again, we expect these to vary depending on both the size and distribution of the data. We will determine bounds in terms of the computational measures of centralized data mining algorithms performing the same task.

Security provided. It is rarely possible to guarantee perfect security. Encryption can be broken, outside knowledge added to information passed by the algorithms may reveal individual values, etc. Each problem faces different security requirements. For a solution that meets these requirements, we will evaluate the difficulty of breaching security. Examples might be "finding what rules are supported by a site is as hard as breaking DES", or "if values for 1/2 the data are known, the others are revealed".

Another key measure of success of this research will be the ease of addressing a new real-world problem: Will these efforts require considerable effort from the PI? Or will we succeed in developing a methodology enabling new problems to be addressed as undergraduate research projects? This will largely be measured through outreach to undergraduate and master's students both in the classroom and through research. In particular, we plan to seek Research Experiences for Undergraduates supplemental funding to explore our ability to make this technology widely accessible.

5 Broader Impacts

This proposed project will be conducted by PIs from two different information assurance centers: CERIAS of Purdue University and SAI (Systems Assurance Institute) of Syracuse University. The Center for Education and Research in Information Assurance and Security, or CERIAS, is a foremost University center for multidisciplinary research and education in areas of information security. Our areas of research include computer, network, and communications security as well as information assurance. The center's mission is to establish an ongoing center of excellence which will promote and enable world class leadership in multidisciplinary approaches to information assurance and security research and education. SAI is one of the affiliates of CERIAS whose affiliate program aims at helping leverage resources and provide community support for groups performing graduate research-based education in information security. The collaboration of this project will enhance the relationship between SAI and CERIAS, enhance sharing of information and research results, and provide a wider variety of education and training opportunities for students.

This project will be integrated in the CERIAS ongoing seminar series, available both to students and to external professionals via live webcast. The investigators have already taught privacy preserving data mining in such diverse forums as a half day tutorial at the 2002 *European Conference on Machine Learning* and *Conference on Principles and Practice of of Knowledge Discovery and Data Mining*, the CERIAS seminar series, and the Purdue School of Science Freshman Honor's seminar. Such educational outreach will continue and evolve as new technology is developed under the project.

The direct educational impact of this project will be two-fold. First, by recruiting and involving undergraduate and graduate students, especially those belonging to underrepresented groups, in various facets of the project, we will contribute directly to their professional development. Second, by providing technologies for working with otherwise restricted information, we aim to develop an invaluable instructional tool.

While the proposed research is specifically in information technology, interaction with professionals in other disciplines will focus the problems to be addressed. Particular emphasis will be placed on areas where the analysis of large data sets has high potential impact, but privacy concerns limit the ability to acquire such data sets (e.g., medical research.) This will lead to a multidisciplinary impact, enabling information technology to support research that otherwise would be difficult or impossible.

References

[1] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Santa Barbara, California, USA: ACM, May 21-23 2001, pp. 247–255. [Online]. Available: http://doi.acm.org/10.1145/375551.375602

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases*. Santiago, Chile: VLDB, Sept. 12-15 1994, pp. 487–499. [Online]. Available: http://www.vldb.org/dblp/db/conf/vldb/vldb94-487.html

[3] ——, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD Conference on Management of Data*. Dallas, TX: ACM, May 14-19 2000, pp. 439–450. [Online]. Available: http://doi.acm.org/10.1145/342009.335438

[4] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, "Disclosure limitation of sensitive rules," in *Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, Chicago, Illinois, Nov. 8 1999, pp. 25–32. [Online]. Available: http://ieeexplore.ieee.org/iel5/6764/18077/00836532.pdf?isNumber=18077&%prod=CNF&arnumber=00836532

[5] M. J. Atallah and W. Du, "Secure multi-party computational geometry," in *Seventh International Workshop on Algorithms and Data Structures (WADS 2001)*, Providence, Rhode Island, USA, Aug. 8-10 2001. [Online]. Available: http://www.cerias.purdue.edu/homes/duw/research/paper/wads2001.ps

[6] D. Beaver, "Server-assisted cryptography," in *Proceedings of the 1998 New Security Paradigms Workshop*, Charlottesville, VA USA, September 22-26 1998.

[7] ——, "Commodity-based cryptography (extended abstract)," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. El Paso, Texas, United States: ACM Press, 1997, pp. 446–455.

[8] P. Chan, "An extensible meta-learning approach for scalable and accurate inductive learning," Ph.D. dissertation, Department of Computer Science, Columbia University, New York, NY, 1996. [Online]. Available: http://www.cs.columbia.edu/~pkc/papers/thesis.ps

[9] ——, "On the accuracy of meta-learning for scalable data mining," *Journal of Intelligent Information Systems*, vol. 8, pp. 5–28, 1997. [Online]. Available: http://www.cs.columbia.edu/~sal/hpapers/jiis.ps.gz

[10] D. W.-L. Cheung, J. Han, V. Ng, A. W.-C. Fu, and Y. Fu, "A fast distributed algorithm for mining association rules," in *Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS'96)*. Miami Beach, Florida, USA: IEEE, Dec. 1996, pp. 31–42.

[11] D. W.-L. Cheung, V. Ng, A. W.-C. Fu, and Y. Fu, "Efficient mining of association rules in distributed databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 911–922, Dec. 1996.

[12] C. Clifton, "Using sample size to limit exposure to data mining," *Journal of Computer Security*, vol. 8, no. 4, pp. 281–307, Nov. 2000. [Online]. Available: http://iospress.metapress.com/openurl.asp?genre=article&issn=0926-227X&%volume=8&issue=4&spage=281

[13] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu, "Tools for privacy preserving distributed data mining," *SIGKDD Explorations*, vol. 4, no. 2, pp. 28–34, Jan. 2003. [Online]. Available: http://www.acm.org/sigs/sigkdd/explorations/issue4-2/contents.htm

[14] C. Clifton, M. Kantarcioglu, and J. Vaidya, “Defining privacy for data mining,” in *National Science Foundation Workshop on Next Generation Data Mining*, H. Kargupta, A. Joshi, and K. Sivakumar, Eds., Baltimore, MD, Nov. 1-3 2002, pp. 126–133.

[15] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang, “Finding interesting associations without support pruning,” in *Proceedings of the 16th International Conference on Data Engineering*, San Diego, California, Feb. 28 – Mar. 3 2000. [Online]. Available: http://www.computer.org/proceedings/icde/0506/05060489abs.htm

[16] G. Di-Crescenzo, Y. Ishai, and R. Ostrovsky, “Universal service-providers for database private information retrieval,” in *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*, September 21 1998.

[17] W. Du, “A study of several specific secure two-party computation problems,” Ph.D. dissertation, Purdue University, West Lafayette, Indiana, 2001. [Online]. Available: http://www.cerias.purdue.edu/homes/duw/research/duthesis.pdf

[18] W. Du and M. J. Atallah, “Protocols for secure remote database access with approximate matching,” in *7th ACM Conference on Computer and Communications Security (ACMCCS 2000), The First Workshop on Security and Privacy in E-Commerce*, Athens, Greece, November 1-4 2000. [Online]. Available: http://portal.acm.org

[19] ——, “Privacy-preserving cooperative scientific computations,” in *14th IEEE Computer Security Foundations Workshop*, Nova Scotia, Canada, June 11-13 2001, pp. 273–282. [Online]. Available: http://portal.acm.org

[20] ——, “Privacy-preserving statistical analysis,” in *Proceeding of the 17th Annual Computer Security Applications Conference*, New Orleans, Louisiana, USA, December 10-14 2001. [Online]. Available: http://www.cerias.purdue.edu/homes/duw/research/paper/acsac2001.ps

[21] W. Du and Z. Zhan, “Building decision tree classifier on private data,” in *IEEE International Conference on Data Mining Workshop on Privacy, Security, and Data Mining*, C. Clifton and V. Estivill-Castro, Eds., vol. 14. Maebashi City, Japan: Australian Computer Society, Dec. 9 2002, pp. 1–8. [Online]. Available: http://crpit.com/Vol14.html

[22] “Directive 95/46/EC of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data,” *Official Journal of the European Communities*, vol. No I., no. 281, pp. 31–50, Oct. 24 1995. [Online]. Available: http://europa.eu.int/comm/internal_market/privacy/

[23] A. Eisenberg, “With false numbers, data crunchers try to mine the truth,” *New York Times*, July 18 2002. [Online]. Available: http://query.nytimes.com/search/abstract?res=F10C14F6395D0C7B8DDDAE0894%DA404482

[24] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy preserving mining of association rules,” in *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 23-26 2002, pp. 217–228. [Online]. Available: http://doi.acm.org/10.1145/775047.775080

[25] O. Goldreich, S. Micali, and A. Wigderson, “How to play any mental game - a completeness theorem for protocols with honest majority,” in *19th ACM Symposium on the Theory of Computing*, 1987, pp. 218–229. [Online]. Available: http://doi.acm.org/10.1145/28395.28420

[26] M. Hamblen, “Privacy algorithms: Technology-based protections could make personal data impersonal,” *Computerworld*, Oct. 14 2002. [Online]. Available: http://www.computerworld.com/securitytopics/security/privacy/story/0,10%801,75008,00.html

[27] “Indiana telemedicine incubator,” 2000. [Online]. Available: http://portals.cs.purdue.edu/icds/iti

[28] M. Kantarcioglu and C. Clifton, “Privacy-preserving distributed mining of association rules on horizontally partitioned data,” in *The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD’02)*, Madison, Wisconsin, June 2 2002, pp. 24–31. [Online]. Available: http://www.bell-labs.com/user/minos/DMKD02/Papers/kantarcioglu.pdf

[29] H. Kargupta and P. Chan, Eds., *Advances in Distributed and Parallel Knowledge Discovery.* AAAI/MIT Press, 2000.

[30] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson, “Distributed clustering using collective principal component analysis,” *Knowledge and Information Systems*, vol. 3, no. 4, pp. 405–421, Nov. 2001.

[31] J. Li, X. Zhang, G. Dong, K. Ramamohanarao, and Q. Sun, “Efficient mining of high confidence association rules without support thresholds,” in *3rd European Conference on Principles and Practice of Knowledge Discovery in Databases*, Prague, Czech Republic, Sept. 15–18 1999.

[32] Y. Lindell and B. Pinkas, “Privacy preserving data mining,” in *Advances in Cryptology – CRYPTO 2000.* Springer-Verlag, Aug. 20-24 2000, pp. 36–54. [Online]. Available: http://link.springer.de/link/service/series/0558/bibs/1880/18800036.htm

[33] K. Muralidhar, R. Sarathy, and R. A. Parsa, “A general additive perturbation method for database security,” *Management Science*, vol. 45, no. 10, pp. 1399–1415, 1999.

[34] S. C. Pohlig and M. E. Hellman, “An improved algorithm for computing logarithms over GF(p) and its cryptographic significance,” *IEEE Transactions on Information Theory*, vol. IT-24, pp. 106–110, 1978.

[35] A. Prodromidis, P. Chan, and S. Stolfo, *Advances in Distributed and Parallel Knowledge Discovery.* AAAI/MIT Press, 2000, ch. 3: Meta-learning in distributed data mining systems: Issues and approaches. [Online]. Available: http://www.cs.columbia.edu/~andreas/publications/DDMBOOK.ps.gz

[36] A. L. Prodromidis and S. J. Stolfo, “Cost complexity-based pruning of ensemble classifiers,” *Knowledge and Information Systems*, vol. 3, no. 4, pp. 449–469, Nov. 2001.

[37] S. J. Rizvi and J. R. Haritsa, “Maintaining data privacy in association rule mining,” in *Proceedings of 28th International Conference on Very Large Data Bases.* Hong Kong: VLDB, Aug. 20-23 2002, pp. 682–693. [Online]. Available: http://www.vldb.org/conf/2002/S19P03.pdf

[38] P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression,” in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, May 1998.

[39] A. Savasere, E. Omiecinski, and S. B. Navathe, “An efficient algorithm for mining association rules in large databases,” in *Proceedings of 21st International Conference on Very Large Data Bases.* VLDB, Sept. 11-15 1995, pp. 432–444. [Online]. Available: http://www.vldb.org/dblp/db/conf/vldb/SavasereON95.html

[40] Y. Saygin, V. S. Verykios, and C. Clifton, “Using unknowns to prevent discovery of association rules,” *SIGMOD Record*, vol. 30, no. 4, pp. 45–54, Dec. 2001. [Online]. Available: http://www.acm.org/sigmod/record/issues/0112/SPECIAL/5.pdf

[41] P. Scheuermann, “Distributed web log mining using maximal large itemsets,” *Knowledge and Information Systems*, vol. 3, no. 4, pp. 389–404, Nov. 2001.

[42] B. Schneier, *Applied Cryptography*, 2nd ed. John Wiley & Sons, 1995.

[43] D. B. Skillicorn and Y. Wang, “Paralle and sequential algorithms for data mining using inductive logic,” *Knowledge and Information Systems*, vol. 3, no. 4, pp. 405–421, Nov. 2001.

[44] J. Vaidya and C. Clifton, “Privacy preserving association rule mining in vertically partitioned data,” in *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 23-26 2002, pp. 639–644. [Online]. Available: http://doi.acm.org/10.1145/775047.775142

[45] R. Wirth, M. Borth, and J. Hipp, “When distribution is part of the semantics: A new problem class for distributed knowledge discovery,” in *Ubiquitous Data Mining for Mobile and Distributed Environments workshop associated with the Joint 12th European Conference on Machine Learning (ECML’01) and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD’01)*, Freiburg, Germany, Sept. 3-7 2001. [Online]. Available: http://www.cs.umbc.edu/~hillol/pkdd2001/papers/wirth.pdf

[46] A. C. Yao, “How to generate and exchange secrets,” in *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*. IEEE, 1986, pp. 162–167.

[47] M. J. Zaki, “Parallel and distributed association mining: A survey,” *IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining*, vol. 7, no. 4, pp. 14–25, Dec. 1999.