

# PBKM: A Secure Knowledge Management Framework

(extended abstract)

Shouhuai Xu and Weining Zhang

Department of Computer Science, University of Texas at San Antonio  
{shxu,wzhang}@cs.utsa.edu

## Abstract

*In this paper, we advocate research into techniques that support the extraction, sharing, and utilization of knowledge for collaborative problem solving applications. We present a system framework for secure knowledge management, called PBKM, which in addition to providing standard security mechanisms such as access control, will possess three crucial features, namely, privacy-preservation which should be ensured in a knowledge-extraction procedure, breaching-awareness which should be taken into consideration in the knowledge-dissemination procedure, and abuse-accountability which is incorporated in the management of knowledge. We explore this framework by elaborating on its components and their relationship to existing techniques such as database, cryptography, data mining, and machine learning. We identify a number of challenging issues and interesting problems for further research.*

**Keywords:** knowledge management, knowledge extraction, knowledge breaching, abuse accountability, security, privacy

## 1. Introduction

### 1.1. Motivation

The advancement in networking, storage, and processor technologies has brought in an unprecedented amount of digitalized information. In order to effectively utilize collected data in applications, organizations routinely use Database Management Systems (DBMSs) to store, manage, and use the collected data. While it is well-accepted that data has become vital assets of organizations, what many decision-making applications really need is the knowledge hidden in the raw data. For this reason, knowledge-extraction technologies such as data mining and machine learning have been developed in recent years to make it feasible to "refine" large volumes

of raw data into succinct knowledge that can be directly utilized in decision-making applications. However, most current data mining based applications are designed to solve problems for the owners of the data, that is, the data mining is performed on the data of an organization to solve business problems of the same organization. Although still very popular, such use of data mining is limited and needs to be extended.

As the Internet quickly evolves into a global computational infrastructure, it provides a platform for new applications that allow autonomous organizations to collaboratively solve problems using data mining. Although collaborating parties may perform data mining by directly sharing their data, which is indeed a common scenario in current practice, a direct data sharing is often limited or not feasible due to a number of reasons such as the data sharing policies of organizations, the concern of privacy protection, the technical difficulties in dealing with heterogeneity and the sheer volume of the data, just name a few. On the other hand, since knowledge extracted from the data is often more abstract and less bulky than the raw data, the sharing of extracted knowledge may be much easier and more beneficial than sharing of data in many problem solving scenarios. Thus, for such applications, there is a growing need for the extracted knowledge to be shared among collaborating parties.

One desirable feature of this "knowledge-sharing" paradigm is the distinction between the *knowledge-extraction* process in which data mining algorithms are applied to discover the hidden knowledge in data, and the *knowledge-dissemination* process in which the discovered knowledge is utilized in applications to solve problems. Such distinction is natural and sometimes necessary for collaborative problem solving applications that utilize a number of emerging technologies such as software agents, web services, and XML databases. For example, in an application involving collaborating software agents of different organizations, it is natural to have an agent to collect data into a database, a second agent to extract knowledge from the collected data and to store the

result in a knowledge base, a third agent to use the extracted knowledge to make a decision, and a fourth agent to act on the decision. A similar scenario can also occur in the service-oriented computing paradigm.

Together, the needs of knowledge sharing and the distinction between knowledge extraction and knowledge dissemination raise many issues regarding the creation, the management, and the utilization of extracted knowledge. To address these issues, there is a need for a flexible *framework* for Knowledge Management Systems (KMSs) that provide the basic and common functionalities required to effectively coordinate the knowledge extraction with the knowledge dissemination. In this paper, we present such a framework with an emphasis on security and privacy protection. To the best of our knowledge, such a framework has not been proposed in the literature.

## 1.2. Our Contributions

Our main contribution in this paper is two-fold:

1. We argue for the need of a systematic investigation on Knowledge Management Systems. We envision that a KMS has the functionalities in the flavor of the traditional Database Management Systems (DBMSs): (1) It facilitates the extraction of knowledge from existing traditional database and/or knowledge-base systems. The extraction of knowledge may be based on some data mining algorithms. (2) It facilitates storage, retrieval, integration, transformation, visualization, and analysis of extracted knowledge structures (e.g., decision trees, association rules, neural networks). Besides, it also supports construction of new knowledge structures from those existing ones to form knowledge that is deeper than the knowledge directly extracted from the raw data. (3) It facilitates the utilization of the managed knowledge. The utilization may be simply to give the knowledge to a knowledge consumer, or to answer the knowledge consumers' queries through a web service.
2. Besides traditional security goals such as authorization, authentication, and access control, we specify two new security goals of secure knowledge management systems, namely *privacy-preservation* and *breaching-awareness*, where *privacy-preservation* means that the knowledge extraction process should not compromise the privacy of the source data, and *breaching-awareness* means that a system policy regarding knowledge dissemination must take into account the seemingly inevitable knowledge breaching. While these two properties are common to most knowledge management systems, we introduce another property, called *abuse-accountability*,

which is also crucial to many knowledge management systems. The motivation is that abuse of knowledge could result in much more catastrophic consequences than abuse of data, and thus we need technical means to hold the abuser accountable. Thereby, we present and explore a framework for secure knowledge management called Privacy-preserving and Breaching-aware Knowledge Management (PBKM).

## 1.3. Organization

The rest of this paper is organized as follows. In Section 2, we present our privacy-preserving and breaching-aware knowledge management system framework. In Section 3, we present three example scenarios that demonstrate the flexibility of the proposed framework. In Section 4, we discuss work related to ours. In Section 5, we describe a number of challenges and our on-going research work.

## 2. The PBKM Framework

The PBKM is a conceptual role-based system framework of a knowledge management system in which loosely-coupled autonomous software systems, called the participating systems or the parties, collaborate in extraction and dissemination of knowledge. Throughout this paper, we use the term *knowledge* to refer to knowledge models, such as decision trees, association rules, or neural networks, that are extracted from raw data and expressed in a representation language, such as extensible markup language (XML) [7]. By *knowledge management* we mean the methodology for systematically extracting and utilizing of the knowledge. A *Knowledge Management System* (KMS) is a collection of collaborative software systems that collectively provide the functionality needed to perform the tasks of knowledge management. The purpose of PBKM framework is to define various roles that are played by participating systems, the relationships among different roles, and how they are related to the two key functionalities of a KMS, namely knowledge-extraction and knowledge-dissemination, and the security goals. In this Section, we present the PBKM framework by specifying a system model, the adversary, and security goals.

### 2.1. A Model

As shown in Figure 1, at the heart of the PBKM is a Knowledge Management System (KMS), which can be thought of abstractly as a system that takes data and rules as input, extracts knowledge from the data (possibly with the help of the input rules), manages the extracted knowledge,

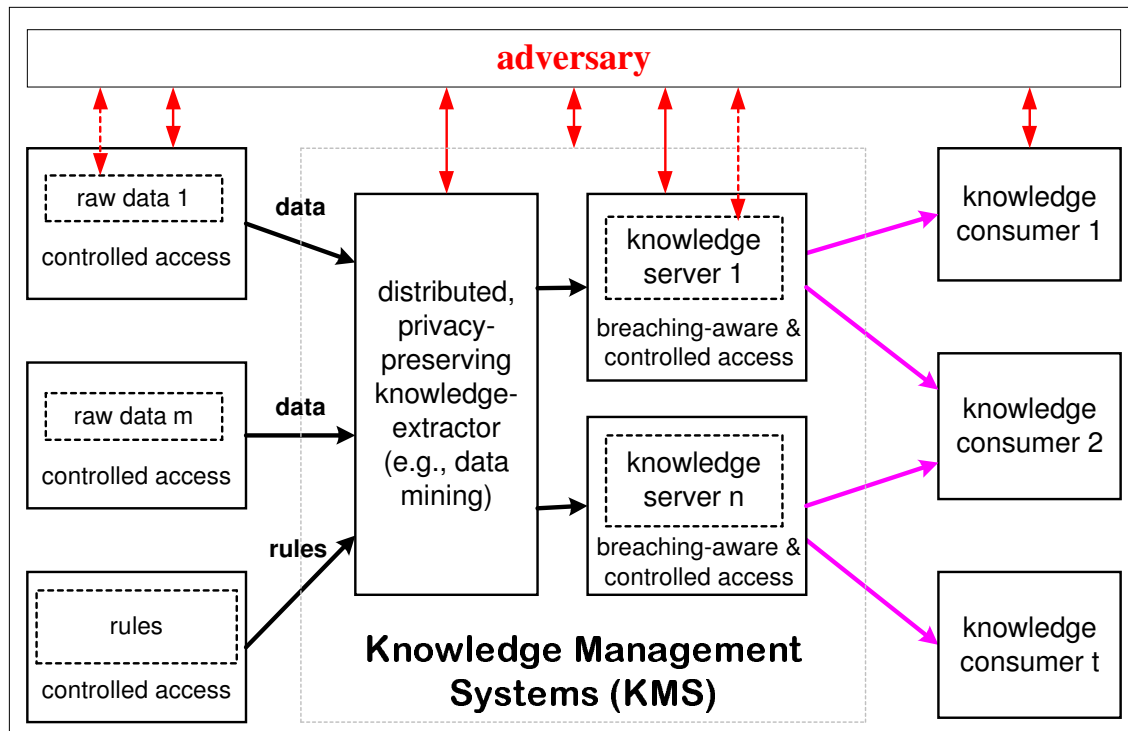


Figure 1. The PBKM Framework

and provides knowledge based services to knowledge consumers. The inputs of the KMS are taken from data sources (for data) and rules bases (for rules) that provide controlled access to their content. The hidden knowledge is extracted by knowledge extractors and disseminated through knowledge servers to knowledge consumers. All components of a PBKM are subjected to attacks of an adversary. In this framework, all components as well as the adversary are roles played by the participating systems (which are assumed to be autonomous computational artifact, such as, computer programs of different organizations). It is important to notice that in an instance of the framework a party may play multiple roles and one of the roles may be the adversary. For example, a party may have a database and need to use the knowledge extracted from its own database as well as from databases owned by other parties. In this case, the party plays the roles of a data source and a knowledge consumer, but not as other roles. It is understood that the knowledge needed by this party will be extracted and disseminated by other parties of the system while they are playing their respective roles defined in the PBKM framework. If in addition this party also participates in a multi-party data mining activity, it will also play the role of a knowledge extractor.

In the following, we explore the PBKM framework by describing the roles and their relationships in more details.

Although an instance of PBKM could be a centralized system (see Section 3 for more details), we believe that most useful instances are typically distributed systems. Without loss generality, we explore the model with an emphasis on its distributed nature.

**Input to KMS** The input to a KMS is a set of *datasets* and optionally a set of *rules* from databases and rule-bases, respectively, where a *dataset* may contain any type of data, such as, structured data from database or data warehouse systems, text/multimedia documents from information repositories, or web pages, and the *rules* may be expressed in various formats, such as the production rules found in a typical expert system, or derivation rules in other rule-based systems. The input rules may be extracted through a knowledge engineering process or be learned automatically. These rules may be used by knowledge extractors in various stages of knowledge extraction, for example, in a rule-based information extraction process during the preparation of datasets for mining. In Fig. 1, the input to KMS include  $m$  raw datasets and one set of rules.

We stress that the access to the datasets and rules are protected by their respective sources through appropriate security policies (e.g., Mandatory Access Control, Discretionary Access Control, Role-Based

Access Control), and that the *controlled access* may be enforced, for instance, by a security mechanism implemented in a DBMS. Moreover, the datasets and rules may be owned by different parties that are presumably prohibited from sharing, or not willing to share, their datasets/rules, although they are allowed to take advantage of the data in their own decision-making applications.knowledge

**KMS** From a functionality perspective, a KMS is analogous to a traditional Database Management System (DBMS). However, there are some fundamental differences: (1) The objects managed by a KMS are knowledge models such as decision trees or association rules. Whereas, the objects managed by a DBMS are raw data. (2) Parties are autonomous and a party may play one or more roles.<sup>1</sup> (3) A KMS is strictly more powerful than a DBMS, because it must ensure two properties, namely *privacy-preserving knowledge extraction* and *breaching-aware knowledge dissemination*. Whereas, no such requirement is specified in a traditional DBMS.

The KMS consists of components that play three types of roles: *knowledge extractor*, *knowledge server*, and *knowledge manager*. For example, in the specific instance of KMS in Fig. 1 there are one knowledge extractor and  $n$  knowledge servers, and each one of them also is a knowledge manager. The functionality of these three roles are as follows.

- **Knowledge Extractor.** A knowledge extractor provides supports for knowledge extraction tasks which for example may include the preparation of data, the specification of extraction tasks, and the execution of extraction algorithms. A knowledge extractor may be fully automated or interactive. Knowledge can be extracted from datasets owned by different owners (that is, parties) using an appropriate method such as distributed data mining. A key feature of knowledge extractors is that they must guarantee that the extraction of knowledge will not compromise individual privacy. This feature can be ensured by the so-called *privacy-preserving data mining* techniques (see 2.3.1).
- **Knowledge Server.** A knowledge server provides services to knowledge consumers. The simplest form of the service is to deliver an extracted knowledge model to a knowledge consumer. However, more sophisticated, and value-added services may require a non-trivial utilization of extracted knowledge. For example, a knowledge server may provide a service by using the extracted knowledge to answer queries posted

by a decision-making application of a knowledge consumer. Such services may be implemented through a variety of techniques, such as web services and software agents. A key feature of knowledge servers is that they are *breaching-aware* (see 2.3.2).

- **Knowledge Manager.** A knowledge manager provides supports for storage, retrieval, analysis, integration, visualization, and transformation of extracted knowledge. In other words, a knowledge manager is to knowledge what a database management system is to data. In a KMS, knowledge managers are often not separable from other roles of the system since they provides a set of functionality that is fundamental to both knowledge extractors and knowledge servers. Extracted knowledge may be expressed in various representation languages, for example, a particularly useful representation language may be the XML-based Predictive Model Markup Language (PMML) proposed by the Data Mining Group as a data mining standard. With extracted knowledge represented as XML documents, the emerging XML database management systems can be leveraged to implement knowledge managers.

**Output of KMS** The KMS disseminates knowledge to knowledge consumers through an appropriate interface (e.g., web services). For example, a knowledge consumer may ask one or more knowledge servers certain questions, so that the answer(s) will be utilized in the knowledge consumer's decision making procedure. The access to the knowledge may be controlled via an appropriate policy, and enforced via an appropriate system.

## 2.2. Adversary

An adversary is a software entity that may perform a variety of activities to compromise the system. In PBKM framework, an adversary may interact with any component of the KMS in various ways, including collusion with other adversaries in a coordinated attack. Specifically, we consider the following activities of an adversary.

- Besides having legitimate access to a data source or a rule base through the controlled access interfaces (e.g., authorized queries to a database), the adversary may have unauthorized access to some data or rules, perhaps through the underlying system components (e.g., operating systems). In an extreme case, the adversary may have completely corrupted one or more of the data sources and rule bases.
- The adversary knows the internal structure of the KMS. For example, it knows how the extracted knowl-

---

<sup>1</sup> This may seem anti-intuitive, but cryptographic techniques do facilitate it.

edge are organized and stored, and which knowledge extraction algorithms are utilized.

- The adversary may have corrupted a subset of parties in a distributed privacy-preserving knowledge-extraction procedure. We further elaborate on this below.
- The adversary may have access to one or more knowledge servers via interfaces that are different from those available to knowledge consumers. Moreover, the adversary may even be able to bypass any provided interface to directly access the knowledge on a knowledge server.
- The adversary may have corrupted one or more knowledge consumers. As a consequence, the queries presented by a corrupted knowledge consumer may speed up the breaching of the targeted knowledge stored at certain knowledge servers.

By being aware of what an adversary may do to the system, the system can be protected from the attacks of the adversary. Such a protection must be designed based on the security requirements of the system, and be provided by appropriate techniques at all levels of the system.

### 2.3. New Security Requirements

Besides traditional security requirements such as access control, authorization, and authentication, a KMS should satisfy three new security requirements: privacy-preservation, breaching-awareness and abuse-accountability. By privacy-preservation we mean that the knowledge-extraction procedure must protect individual privacy in the input datasets. By breaching-awareness we mean that the knowledge-dissemination procedure must take into consideration the accumulative leakage of the knowledge used by knowledge servers.

**2.3.1. Privacy-Preserving Knowledge Extraction** We explore it by adopting a *cryptographic secure multi-party computation* approach. Suppose the knowledge extraction procedure involves  $\ell$  parties  $P_1, \dots, P_\ell$  that need to jointly extract knowledge from the input. Further, suppose that party  $P_i \in \{P_1, \dots, P_m\}$  ( $m \leq \ell$ ) has its private input dataset or rule set  $x_i$ , and that party  $P_j \in \{P_{m+1}, \dots, P_\ell\}$  has its input  $x_j = \perp$  (i.e., null). Let  $f : \{x_1, \dots, x_\ell\} \mapsto \{k_1, \dots, k_\ell\}$  be the knowledge extraction function, where  $k_i$  ( $1 \leq i \leq \ell$ ) is the private output (i.e., knowledge in a certain representation language) to party  $P_i$  (including  $k_i = \perp$ ). Informally, by *privacy-preserving knowledge extraction* we mean that there is no adversary  $\mathcal{A}$  that has corrupted a subset of parties  $\Delta \subset \{P_1, \dots, P_\ell\}$  can learn any information about  $x_i$ , where  $P_i \notin \Delta$ , more than what is implied by the function  $f$  as well as the outputs  $y_j$  and the inputs  $x_j$  for

$P_j \in \Delta$ . We refer the reader to [11] for a formal treatment of this notion.

**2.3.2. Breaching-Aware Knowledge Dissemination** A *knowledge breaching* occurs in the knowledge dissemination procedure when an adversary learns the knowledge underlying a knowledge service (that is a service provided by a knowledge server) through a legitimate access to the service. Exactly what constitutes a knowledge breaching will depend on the type of knowledge service (also the type of underlying knowledge). Without loss of generality, we define a knowledge service as a function  $f_K : Q \rightarrow R$ , that maps a (possibly infinite) set of service requests  $Q$ , to a finite set of service responses  $R$  using the underlying knowledge  $K$ . Both the service requests and service responses may be complex data objects, and the mappings from service requests to service responses are defined according to the underlying knowledge. For example, for a knowledge service that classifies job applicants based on existing employees of a company, a service request can be a data record describing a job applicant; the underlying knowledge may be a decision tree learned from the employee database of the company and classifying the employees into three categories: excellent, good, and fair; and the service response can be the category of the applicant predicted by the decision tree.

**Definition 2.1** A *breaching of a knowledge  $K$* , called the target knowledge, occurs if the adversary is able to derive a knowledge  $K'$ , called the learned knowledge, based on a sequence of service requests to and service response from the knowledge service  $f_K$  and define a knowledge service  $f_{K'} : Q \rightarrow R$  using  $K'$ , called the learned service, such that,  $f_K$  and  $f_{K'}$  are indistinguishable.

Notice that, the learned knowledge  $K'$  needs not to be the same as the target knowledge  $K$ . We assume that the adversary is an legitimate knowledge customer of the service, thus she is able to issue a sequence of service requests and use the corresponding service responses to derive  $K'$ . Also notice that the knowledge breaching is relevant only to knowledge services that do not explicitly disclose the target knowledge  $K$ . On the other hand, since each pair of service request and service response disclose to the adversary some information about  $K$ , a knowledge breaching is gradual and seemingly inevitable given unlimited access to the knowledge service.

The KMS must be aware of possible knowledge breaching and deal with it through appropriate knowledge dissemination policies. We stress that a KMS is required to be breaching-aware rather than breaching-proof because the ultimate goal of the knowledge dissemination is to provide knowledge services to knowledge consumers using the extracted knowledge, and the requirement of breaching-proof

may severely limit the types of services that can be provided by the KMS. In PBKM, we view breaching-awareness as an requirement for the KMS to make appropriate knowledge dissemination policies, such as how the knowledge extractors and knowledge servers should be compensated.

Since a knowledge breaching is a gradual process, at any given point of this process, the learned knowledge may be incomplete, therefore will likely to cause the learned service to respond to a service request differently from the given knowledge service. As a requirement of the breaching-awareness, it makes sense to measure the degree of knowledge breaching, which is captured by the following definition.

**Definition 2.2** Let  $f_K$  be a knowledge service with a target knowledge  $K$  and  $f_{K'}$  be a service defined according to a learned knowledge  $K'$  derived by the adversary, we say that  $K'$  causes a degree  $\sigma$  knowledge breaching of  $K$  at significance level  $\alpha$ , if the adversary is able to derive  $K'$ , so that  $Pr(f_{K'}(\cdot) = f_K(\cdot)) > \sigma$  with a probability  $1 - \alpha$ , where  $0 \leq \sigma \leq 1$ ,  $0 \leq \alpha \leq 1$ , and  $Pr(f_{K'}(\cdot) = f_K(\cdot))$  is the probability that the two services give the same response to a service request.

The KMS needs to provide support to model and to measure the degree of knowledge breaching for various types of knowledge services.

**2.3.3. Abuse-Accountability** Abuse of knowledge could result in more catastrophic consequences than abusing of data. So we need technical means to hold those abusers accountable. This may be crucial to certain knowledge management systems (e.g., the systems coordinating government agencies' counter-terror activities).

### 3. Three Example Scenarios

In the following, we demonstrate the generality of the PBKM framework by describing three specific instances of the framework.

#### 3.1. A Simple Case

In the simplest case, we only consider the setting where there is a single organization that owns the data, buys a data mining software, and runs the software to extract knowledge that will be exclusively used by the organization itself. In this case, there is perhaps no privacy-preservation and breaching-awareness issues. However, there may still be issues of abuse-accountability, because the abuse of the knowledge can result in significant consequences.

As a further step towards what we called KMS, the organization may not have to buy the data mining software. Instead, it can use that software through "application as a ser-

vice". In this case, the issues of privacy-preserving emerge: the application server (i.e., data mining software owner) should not learn any information about the organization's datasets, while allowing the organization to obtain the extracted knowledge. In principle, cryptographic multi-party computation can solve this problem.

#### 3.2. A Business Case

We now consider a scenario arising in an emerging computing paradigm called "knowledge-as-a-service" [22], which is a natural extension of service oriented computing, such as "application as a service" and "database as a service" [13]. These service oriented paradigms emerge as cost-effective business models in response to increasing business competition and to the cost of keeping the desired computational, data management, and knowledge discovery capabilities that has become too high to be justified for many organizations. By delegating computational, data management, and knowledge discovery tasks to appropriate service providers, organizations can better satisfy their information processing needs with much lower costs. The "knowledge-as-a-service" serves an example of the separation of knowledge extraction and knowledge utilization, and is justified for the following reasons.

1. The rising costs of knowledge extraction. Data mining is a specialized and complex task that involves many steps and requires well trained personnel. Despite the tremendous advance in hardware, software, and networking technologies, the costs associated with knowledge extraction is still on the rise. These costs are for the acquisition of software, hardware, datasets, and the maintenance and management of systems. The situation is further complicated if one needs accurate knowledge and strict privacy in the knowledge extraction procedure (see more discussion below).
2. Restricted access to data. Although knowledge models are often extracted by an organization from its own datasets, much of useful knowledge may be in data owned by other organizations. Access to data of another organization may be prohibited by law or policies. For example, the national criminal databases can only be accessed by law enforcement organizations. Likewise, hospital patients data is only accessible to relevant health-care organizations. Yet another typical scenario is that competition rivals would never share their data, but would benefit from knowledge extracted from each other's datasets. As a consequence, just like that data are valuable assets of today's organizations, knowledge models will be valuable assets of tomorrow's organizations.

3. Limited choice of technology that addresses privacy concerns. The emerging of the data mining industry has inspired a lot of concerns on individual privacy [6, 19]. To relieve these concerns, privacy-preserving data mining techniques have been proposed. Currently, these techniques are still in an early stage of development, and one may have to choose between techniques that are efficient but may generate less accurate knowledge models (e.g., perturbation-based data mining techniques [1, 8]) and techniques that generate accurate knowledge models but are much less efficient (e.g., cryptographic secure multi-party computation based data mining techniques [14, 23, 12]). As a consequence, one who is interested in accurate knowledge and strong privacy guarantee may be forced to conduct computation- and communication-extensive tasks, which may incur significant investment.
4. Different needs of knowledge by different applications. There are many ways that knowledge models can be utilized. Two extreme examples are: (1) An application needs to own an entire knowledge model. (2) An application only needs to apply (rather than to own) a knowledge model to certain instance data. The difference between these two types of utilization is comparable to the difference between buying a car and taking a taxi, or to the difference between purchasing an expensive full-fledged software system and paying only for some of the needed functionality.

As an example of “knowledge-as-a-service”, consider a risk assessment application involving three independent insurance companies, say two automobile insurance companies  $A$  and  $B$ , and a life insurance company  $C$ . In order to determine the life insurance premium of a new client, Bob, so that to minimize the risk of financial loss and maximize the profit,  $C$  would like to know the likelihood of Bob being involved in severe car accidents (e.g., those who drive aggressively would more often cause accidents and should pay more for their life insurance). Clearly, a predictive model learned from the client databases of car insurance companies  $A$  and  $B$  will be useful to  $C$ . However,  $C$  does not have access to the databases of  $A$  or  $B$ , since such accesses would necessarily compromise the client’s privacy. Moreover, given that the datasets of  $A$  and  $B$  are their assets, they would not allow  $X$  to mine their datasets even if it is technically possible. Fortunately, a knowledge service provider  $D$  can use knowledge models extracted from datasets of  $A$  and  $B$  to provide the likelihood information needed by  $C$  and many other insurance companies, perhaps in a way that  $D$  pays the datasets owners,  $A$  and  $B$ , and gets paid off by the knowledge consumers.

In this example, companies  $A$  and  $B$  play individually the role of data source and jointly the role of knowledge

extractor. Company  $C$  on the other hand, plays the role of knowledge consumer. The knowledge server  $D$  provides a simple type of knowledge service, called *classification service*, which classifies a customer-supplied data instance (as a service request) based on a knowledge model such as a decision tree or a neural network. The privacy-preservation may be guaranteed by a privacy-preserving multi-party data mining protocol. The breaching-awareness may be incorporated into the cost structure of the service provided by  $D$  (see [22] for more details).

### 3.3. A Government Case

Suppose multiple government agencies have their own databases for intelligence information. They might not want to completely open their databases to the other agencies, in fear of leaking the information traced back to the sources. However, they need to collaborate in counter-terror.

In this case, abuse-accountability is also be crucial if the abuse of knowledge can result in much more catastrophic consequences (say, than those incurred by the abuse of data). For example, if the knowledge is extracted from terrorists databases and enables the law-enforcement to profile terrorists, then abuse (say, leakage due to an insider attack) of this knowledge may significantly increase the difficulties of the law-enforcement in counter-terror.

## 4. Related Work

### On the evolution of service based computing paradigms.

Service oriented computing is an active research area and a number of service types can be identified, including “application as a service”, “database as a service” [13], “data mining model as a service” [18], and the more general notion of “web service”. The PBKM framework emphasizes on the security issues of the types of services that are based on extracted knowledge models. As a specific instantiation of the above PBKM, we explored the notion of “knowledge as a service” in a setting where a service provider can be compensated [22]. This is particularly relevant in applications such as risk management. For example, a life insurance provider may minimize the risk by determining the premium of a new client based on the likelihood of the client being involved in a fatal car accident, which is a knowledge that a car insurance company could provide. Even in this specific setting, there are many questions left open. Within this framework, we focused on a crucial issue, knowledge breaching, under two specific adversarial strategies: the first one is a new algorithm, and the second is adapted from a known active machine learning algorithm. Through systematic experiments (with various heuristic optimizations), we showed that knowledge breaching is seemingly inevitable. This naturally suggests two counter-strategies: to have the

knowledge provider get paid off (via an appropriate pricing mechanism) and to restrict the number (or types) of queries. We believe that the former is more practical because, after all, the primary driving force behind the emerging of knowledge service is the economic incentives.

**On the relationship to privacy protection of data.** The notion of privacy protection has received tremendous attention in various research communities and contexts. For example, there have been many useful techniques contributed by the cryptography community (cf. [3, 4, 5] and their follow-ons). These techniques target at protecting users' anonymity while allowing them to show their legitimacy. On the other hand, access control protects sensitive data from unauthorized disclosure via direct accesses. However, it cannot prevent indirect accesses. For example, indirect data disclosure via inference channels occurs when sensitive information can be inferred from non-sensitive data and meta-data. We refer the reader to [9] for a survey of inference control in various system settings (e.g., statistical databases, multilevel secure databases, general purpose databases). Very recently, interesting framework and method for eliminating both unauthorized accesses and malicious inferences in the context of OLAP (on-line analytic processing) was investigated [20]. Moreover, a systematic study of the information-disclosure problem in data exchange applications was presented in [15]. We remark that all these techniques don't address the problem of knowledge breaching in the context of knowledge as a service.

**On the relationship to data mining and machine learning.** On one hand, a KMS relies on privacy-preserving data mining techniques to extract knowledge from raw data. Privacy-preserving data mining achieves the goal of preserving the secrecy of individual data records while allowing the derivation of useful patterns. There are two approaches to privacy-preserving data mining. The first approach is to randomize the values in individual records [1]. A model is then built over the randomized data, after first compensating for the randomization (at an aggregate level). This approach is potentially vulnerable to privacy breaches: based on the distribution of the data, one may be able to learn with high confidence that some of the randomized records satisfy a specified property (even though privacy is preserved on average). In general, this approach is still in its early stage (see [8] for the subtleties in capturing the right definition of privacy) and does not provide accurate knowledge. The second approach is based on cryptographic secure multi-party computation techniques [23, 12, 14]. This approach does provide accurate knowledge and a strict privacy guarantee, but is typically much less efficient. In spite of some recent advances in cryptography (e.g., [10]), significant performance improvements are very much needed.

On the other hand, an adversary may achieve a knowledge breaching using data mining and machine learning

techniques. In [22], we investigated, for a special knowledge service called the classification service, a knowledge breaching strategy based on an active machine learning algorithm. Even for this special case, a number of very interesting questions are left open (see next section).

**On the relationship to knowledge sharing.** Knowledge sharing has been recognized a long time ago in the AI community [16] as an important issue in building intelligent systems, where the emphasis was on the sharing of knowledge presented in various knowledge-base and expert systems and on reusing components of system shells. The emerging semantic web techniques [2] continue the quest by incorporating interoperable software agents and shared ontology (which can be viewed as knowledge acquired from human experts) in the scale of the World Wide Web. The sharing of a specific type of knowledge, namely the data mining models, has recently attracted an increasingly more attention of researchers in the database and data mining community [7, 18, 21]. The Predictive Modelling Markup Language [7] has been proposed by the Data Mining Group as a standard format of data mining models. Several popular commercial data mining products as well as a number of research prototype data mining software, such as [21], have already included PMML based import/export capabilities. The general concept of "data mining model as a service" has been proposed in [18]. These efforts are focused either on the mechanisms that enable data mining systems to transfer discovered models to application programs, or on the types of services in which the discovered models can be useful. The PBKM framework described in this paper provides a general system framework that serves as a platform to integrate techniques of knowledge extraction, knowledge sharing, and knowledge utilization in a secure environment. Obviously, the "data mining model as a service" is a special instance of the framework, and the security requirements of PBKM greatly enrich the functionality of knowledge sharing systems.

## 5. Challenges and On-going Works

The PBKM framework raises many interesting research issues and there are many challenges in developing techniques that are necessary for ensuring the privacy-preserving knowledge extraction and breaching-aware knowledge dissemination. In the following, we outline some of the most important issues and challenges.

One challenge is to develop efficient and secure data mining techniques for knowledge extraction. The state-of-the-art privacy-preserving data mining techniques are still in its infancy. As mentioned in Section 4, two lines of research can be identified: cryptographic approaches and database approaches. The cryptographic approaches, represented by

the privacy-preserving multi-party data mining methods, exhibit strong security features and are capable of generating accurate knowledge models. But these techniques suffer from high complexity, thus, are less scalable than database based approaches. On the other hand, database approaches, as represented by perturbation-based methods, exhibit good performance with large datasets, but suffer from deriving less accurate knowledge models.

The breaching-aware knowledge dissemination is a brand new area of research. The biggest challenge in this area is to understand the techniques that might be used by an adversary to breach the knowledge underlying a knowledge service. As we mentioned before, what constitutes a knowledge breaching is dependent of the types of the knowledge service and the type of the underlying knowledge. To this end, the issues raised are much like those in statistical databases where a complete understanding of the inference techniques is a prerequisite for designing a mechanism to protect the data privacy against statistic inference. In [22], we studied a simple form of knowledge breaching where the target knowledge is a classification model (such as a decision tree or a neural network), the learned knowledge is a Boolean valued decision function of a conjunctive form, and the knowledge service is a simple classification service. We considered two methods that may be used by the adversary to derive the learned knowledge, one is based on an active learning method adapted from the machine learning literature, and the other is a heuristic method of our own. Even within this very limited context, there are many directions for further investigations:

- We only considered two breaching strategies that we feel most practical. It is absolutely worthwhile to investigate the behaviors of more strategies, either by designing new methods or by adapting known algorithm (in data mining and machine learning).
- We only considered data domains that have a total order. It is useful to extend the methods to accommodate other data types (e.g., categorical).
- How should an appropriate pricing mechanism (such as a breaching-aware knowledge dissemination policy) be devised and selected? Although a heuristic mechanism can be based on the behaviors of a specific breaching strategy, an optimal mechanism would be based on the minimum number of queries required to extract the targeted knowledge. But which breaching strategy is optimal?
- Initial training samples are relevant in data mining and machine learning based breaching strategies. What is an optimal initial training sample?
- There is an need to develop techniques to measure the degree of knowledge leakage. For example, given

that the number of queries is  $n$  and the significance level is  $\alpha$ , what is the maximum  $\sigma$ , such that the current learned function  $f_{K'}$  represents a degree  $\sigma$  knowledge breaching of  $K$  at level  $\alpha$ ? The answer to this and other similar questions are especially interesting in cases where a collusion of multiple adversaries presents.

- What if a knowledge consumer does not want to leak its instance data to a knowledge server in their query-answer interaction? Although this can be resolved using cryptographic secure function evaluation techniques, they are far from practical. So how can we achieve it practically?
- Since neither of the two known approaches to privacy-preserving data mining is satisfactory, any significant advancement in these techniques can be immediately used in the “knowledge as a service” paradigm.

Obviously, for more general cases of PBKM, there are much more issues related to breaching-awareness.

In addition to security issues, the PBKM framework also raises issues on knowledge improvement, that is the discovery of deep knowledge from discovered knowledge models obtained by different knowledge extractors. Using the knowledge learned from data to directly solve application problems is not the only way to use the learned knowledge, although it is popular in current practice. Most common applications of data mining techniques today are to analyze the data of a single organization and to apply the learned knowledge to solve problems of the same organization. We call such learned knowledge the *local primitive knowledge* because it is often a limited abstraction of the local (the organizational) data viewed at a particular time (the time of the data mining) and biased by a particular analyst (through her choices of mining algorithms, control parameters, training samples and their labeling, etc). In emerging new applications, such as Homeland Security applications, where multiple organizations need to collaborate in decision making processes with restricted access to their local data by other organizations (due to say need-based information sharing policy or privacy protection requirements), local primitive knowledge may be shared (say, through agents or services) among collaborating organizations. A naive approach to the sharing is by way of voting or arbitration based on the decision trees of multiple organizations (again maybe through using agents or services of the organizations). A better approach is to combine the decision trees in a non-trivial way. Even more generally, it is also possible to learn deeper knowledge (at a higher abstraction level) from the local primitive knowledge (which is at a relatively low abstraction level). The techniques developed in a number of data mining research areas such as distributed data

mining and meta-learning [17] are good candidates for being extended to address these issues.

Other issues of the knowledge management include the efficient and effective storage and retrieval of knowledge, visualization and analysis of knowledge, etc. These are much like the corresponding issues in a DBMS, except here the objects to be managed are knowledge rather than data. The need for such functionalities in a KMS is quite obvious. For example, suppose association rules are obtained from mining a number of databases and is stored in the KMS. Due to the privacy-preservation, a knowledge server or a knowledge consumer may not have an access to or any knowledge about the raw data, however, they are able to access to the discovered association rules. In order to understand the similarities, the differences, and other interesting aspects of the raw datasets, tools are needed to analyze, compare, search, and choose these association rules. Although the extracted knowledge can be represented in a number of languages or formats, the current trend is to represent them in XML based standard languages. Thus, we expect the extracted knowledge is a set of XML data and the emerging XML database systems can be leveraged to provide efficient storage and retrieval for KMSs.

## 6. Conclusion

In this paper, we present a system framework for secure knowledge management, which in addition to standard security features, provides three crucial new features, namely, privacy-preservation which is ensured in the knowledge extraction procedure, breaching-awareness which is incorporated in the knowledge dissemination procedure, and abuse-accountability which is provided through knowledge management. We explore the framework by describing the roles played by system components, the relationships among various roles, and how these roles are related to existing technology, such as databases, data mining, privacy preserving, cryptography, etc. We show the generality of the framework by presenting three specific instances of the framework. We identify a number of challenging issues and interesting problems for further research.

## References

- [1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD 2000)*, pages 439–450. ACM, 2000.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
- [3] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24:84–88, Feb. 1981.
- [4] D. Chaum. Blind signatures for untraceable payments. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Advances in Cryptology – CRYPTO 1982*, pages 199–203, New York, 1983. Plenum Press.
- [5] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, Oct. 1985.
- [6] C. Clifton and D. Marks. Security and privacy implications of data mining. In *Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1996.
- [7] Data Mining Group. PMML version 2.1. <http://www.dmg.org>, March 2003.
- [8] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaching in privacy preserving data mining. In *Proceedings of the 2000 Symposium on Principles of Database Systems (PODS 2003)*, pages 211–222. ACM, 2003.
- [9] C. Farkas and S. Jajodia. The inference problem: A survey. *SIGKDD Explorations*, 4(2):6–11, 2003.
- [10] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004.
- [11] O. Goldreich. *The Foundations of Cryptography*, volume 2. Cambridge University Press, 2004.
- [12] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proc. 19th ACM Symp. on Theory of Computing*, pages 218–229. ACM, 1987.
- [13] H. Hacigümüs, S. Mehrotra, and B. Iyer. Providing database as a service. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002)*, pages 29–38. IEEE Computer Society, 2002.
- [14] Y. Lindell and B. Pinkas. Privacy preserving data mining. In M. Bellare, editor, *Advances in Cryptology – Crypto 2000*, pages 36–54. Springer, 2000. *Lecture Notes in Computer Science* No. 1880.
- [15] G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD 2003)*, pages 29–38. ACM, 2004.
- [16] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(30), 1991.
- [17] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 1997.
- [18] S. Sarawagi and S. H. Nagaralu. Data mining models as services on the Internet. *ACM SIGKDD Explorations*, 2(1):24–28, 2000.
- [19] B. Thuraisingham. Data mining, national security, privacy and civil liberties. *SIGKDD Explorations*, 4(2):1–5, 2003.
- [20] L. Wang, S. Jajodia, and D. Wijesekera. Securing OLAP data cubes against privacy breaches. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, page to appear. IEEE Computer Society, 2004.

- [21] D. Wettschereck and S. Muller. Exchanging data mining models with the predictive modelling markup language. In *Proceedings of International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, 2001.
- [22] S. Xu and W. Zhang. Knowledge as a service and knowledge breaching. submitted for publication, 2004.
- [23] A. C. Yao. How to generate and exchange secrets. In *FOCS86*, pages 162–167, Toronto, 1986. IEEE.