

# 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

We envision knowledge management systems as a platform facilitating the extraction, storage, retrieval, integration, transformation, visualization, analysis, dissemination, and utilization of knowledge. We present a system framework for secure knowledge management systems which, in addition to implementing the functionality of knowledge management and providing standard security mechanisms such as access control, possesses three new features: 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 whereby an abuser (typically an insider) or the abusers can be held accountable. 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 also identify a number of challenging 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. To effectively utilize such information, organizations use Database Management Systems (DBMSs). While it is well accepted that data (i.e., raw data, or dataset) has become a vital asset of its owner organization, what is perhaps more important and useful is the

knowledge<sup>1</sup> hidden in the data. For this reason, knowledge-extraction technologies such as data mining and machine learning have been developed to make it feasible to “refine” large volumes of raw data into succinct knowledge that can be directly utilized in decision-making applications. However, knowledge extraction has so far typically been performed on the dataset of a single organization. This paradigm imposes a significant limitation on its usefulness because it does not take into consideration the knowledge hidden in datasets owned by other organizations (even competition rivals), which could be even more useful.

In order for multiple organizations to share the knowledge hidden in their datasets, they can simply put their datasets together and then perform the desired knowledge extraction tasks. However, such a sharing at the data level is likely prohibited for a variety of reasons such as policy/law regulations. A more promising approach is for them to extract knowledge out of the joint dataset while minimizing the exposure of their own datasets to the others; for example, *privacy-preserving multi-party data mining* has been known for some time. In this paper, we move a step forward by exploring a flexible knowledge management<sup>2</sup> framework that not only fulfills a set of necessary functionalities of *multi-party knowledge extraction*, but also deals with various security issues (including privacy-preservation).

### 1.2. Our Contributions

We argue the need for a systematic investigation on Knowledge Management Systems (KMSs). A KMS is a collection of collaborative software components that collectively provide the functionality needed to perform the tasks of knowledge management. We believe that a flexible KMS

---

\* Partially supported by a grant from the Center for Infrastructure Assurance and Security (CIAS), University of Texas at San Antonio.

---

1 In 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 certain knowledge representation language such as the Predictive Model Markup Language (PMML) proposed by the Data Mining Group[5].  
2 In this paper, *knowledge management* means the methodology for systematically extracting and utilizing of knowledge.

should facilitate the distinction between the *knowledge-extraction* processes in which knowledge-extraction algorithms are applied to discover knowledge hidden in the data, and the *knowledge-dissemination* processes whereby the discovered knowledge can be utilized. This is necessary for many applications such as those involving collaborating software agents of different organizations, since it is natural to have an agent collect data into a database, a second agent extract knowledge from the collected data and store the result in a knowledge base, a third agent use the extracted knowledge to make a decision, and yet another agent act on the decision.

We initiate the investigation of a specific system framework for knowledge management systems. The framework is called *Privacy-preserving and Breaching-aware Knowledge Management* (PBKM). We envision a PBKM as an analogy to a DBMS, and should facilitate the following functionalities: (1) the extraction of knowledge from existing database and/or knowledge-base systems using some knowledge extraction (e.g., data mining) algorithms; (2) the storage, retrieval, integration, transformation, visualization, and analysis of extracted knowledge structures (e.g., decision trees, association rules, neural networks); and (3) the utilization of extracted knowledge through dissemination services. Moreover, a PBKM should specify, besides traditional security requirements such as authorization, authentication, and access control, three new security requirements: (1) *privacy-preservation* meaning that the knowledge extraction process should not compromise the privacy of the source data, (2) *breaching-awareness* meaning that a system policy regarding knowledge dissemination must take into account the seemingly inevitable knowledge breaching in the process of knowledge utilization, and (3) *abuse-accountability* meaning that the system should be able to identify and track down the abuse of knowledge so that the abusers can be held accountable.

## 2. The PBKM Framework

PBKM is a system framework of secure knowledge management which enables loosely-coupled autonomous software systems to collaborate for the extraction, storage, dissemination, and utilization of knowledge. In the following, we explore PBKM by specifying the system model, adversary, and security requirements.

### 2.1. A System Model

As shown in Figure 1, at the heart of the PBKM is a Privacy-preserving and Breaching-aware Knowledge Management System (PBKMS), which takes datasets and rules as input, extracts knowledge from the datasets (possibly with the help of the input rules), manages the extracted

knowledge, and provides knowledge to knowledge consumers. All entities involved in this framework including those components inside the PBKMS are autonomous software systems.

Although PBKMS could be a centralized system, we believe that most useful instances are typically distributed. Thus, without loss of generality, we explore the model with an emphasis on its distributed nature.

**Input to PBKMS** An input *dataset* may contain data of any type. For example, an input can be a set of structured data from database or data warehouse systems, text/multimedia documents from information repositories, or web pages. The input *rules* may also be of various types, such as production rules found in typical expert systems or derivation rules in some logic languages. These rules may be extracted through a knowledge engineering or an automatic learning process, and be used by knowledge extractors in the process of knowledge extraction (e.g., a rule-based information extraction during the preparation of datasets for mining). In Figure 1, the input to the PBKMS includes  $m$  datasets and one set of rules.

We stress that access to input datasets and rules are protected through certain security policies (e.g., Mandatory Access Control, Discretionary Access Control, Role-Based Access Control), thereby *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 who are presumably prohibited from sharing, or not willing to share, their datasets/rules, although they are allowed to take advantage of the data for their own decision-making.

**PBKMS** As mentioned before, the functionality of a PBKMS is analogous to that of a secure DBMS. This is so because a PBKMS needs a *knowledge manager* to facilitate the storage and retrieval of knowledge, a *knowledge extractor* to extract knowledge from datasets, and a *knowledge server* to enable the utilization of knowledge. However, there are following fundamental differences. (1) The objects managed by a PBKMS are knowledge models such as decision trees or association rules represented in a suitable language. Whereas, the objects managed by a secure DBMS are raw data. (2) The components of the PBKMS are autonomous and can play multiple roles.<sup>3</sup> Whereas the components of a secure DBMS typically belong to a single owner, and do not play multiple roles, even if they are distributed

---

<sup>3</sup> This may seem anti-intuitive, but cryptographic techniques do facilitate it.

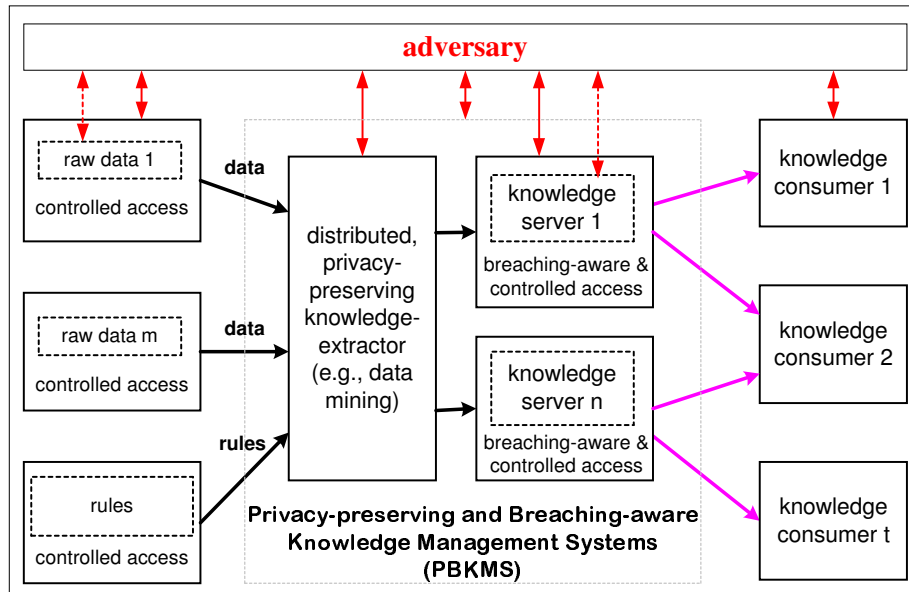


Figure 1. The PBKM Framework

(e.g., a loosely-coupled federated database environment). (3) A PBKMS is strictly more powerful than a secure DBMS, because it must satisfy three additional requirements, namely, *privacy-preserving knowledge extraction*, *breaching-aware knowledge dissemination*, and *abuse-accountability*. Whereas, no such requirement is specified for a traditional secure DBMS. For example, in the specific instance of PBKMS in Figure 1, there are one knowledge extractor and  $n$  knowledge servers, and each one of them has its own knowledge manager that is not explicitly presented (see reason below). Now we elaborate on the functionality of knowledge extractors, knowledge servers, and knowledge managers.

- Knowledge Extractor.** A knowledge extractor supports knowledge extraction tasks which, for example, may include preparation of data, specification of knowledge models to be extracted, and execution of extraction algorithms. A knowledge extractor may be fully automated or interactive. Knowledge can be extracted from datasets owned by different organizations using an appropriate method such as distributed data mining. A key feature of knowledge extractors is that they must guarantee that extraction of knowledge will not compromise individual data. 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 (referred to as knowledge services) to knowl-

edge 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. Such services may be implemented through a variety of techniques, such as web services or 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. We stress that the functionality of knowledge manager might have become native to components of a PBKMS, just like standard data structure has become invisible in the components of a DBMS. This is why we do not explicitly specify knowledge managers in Figure 1. For example, if the extracted knowledge is represented as XML documents, the emerging XML database management systems can be leveraged to implement knowledge managers.

**Output of PBKMS** The PBKMS 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 cer-

tain questions, so that the answer(s) will be utilized in the knowledge consumer’s decision making procedure. We remark that access to the knowledge may be controlled via an appropriate security policy, and enforced via an appropriate security system.

## 2.2. Adversary

In the PBKM framework, an adversary is a probabilistic polynomial-time algorithm, and may interact with any component of the system in various ways as elaborated below.

- 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 PBKMS. For example, it knows how the extracted knowledge 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 in the next subsection.
- 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 have direct access to 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.

## 2.3. Security Requirements

As mentioned before, besides traditional security requirements such as access control, authorization, and authentication, a PBKMS should satisfy three new security requirements: privacy-preservation, breaching-awareness and abuse-accountability.

**2.3.1. Privacy-Preserving Knowledge Extraction** We explore it by adopting a *cryptographic secure multi-party computation* approach [9]. 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, *privacy-preserving knowledge extraction* means 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  $x_i$  or  $k_i$ , where  $P_i \notin \Delta$ , more than what is implied by the function  $f$  as well as the inputs  $x_j$  and the outputs  $k_j$  for  $P_j \in \Delta$ .

**2.3.2. Breaching-Aware Knowledge Dissemination** It may be true that sometimes the knowledge is extracted for an exclusive knowledge consumer, in which case *knowledge breaching* is irrelevant. However, in general many knowledge consumers may obtain a certain partial knowledge from a single knowledge server through an appropriate interface, which is referred to as a *knowledge service*. For example, consider a knowledge service provided by a human-resource consulting firm. The service is based on the knowledge extracted from the employee databases of companies in an industry (this is certainly possible using privacy-preserving knowledge extraction algorithms). Suppose the knowledge is a decision tree that classifies the employees into three categories: excellent, good, and fair. Then, a company can make its decision on hiring by querying the potential performance of its job applicants. In this case, an adversary may indeed be a legitimate knowledge consumer. After making a number of queries, the adversary may be able to derive a knowledge, called the *learned knowledge*, that is *strictly* more than what is conferred by the knowledge servers’ responses. As a consequence, the knowledge underlying the service, called the *target knowledge*, is breached. Worse yet, the *learned knowledge* needs not to be exactly the same form as the *target knowledge*.

Without loss of generality, we define a knowledge service as a function  $f_K : Q \mapsto R$ , that maps a (possibly infinite) set of service requests  $Q$ , to a finite set of service responses  $R$  using the underlying knowledge  $K$ , where both  $Q$  and  $R$  may contain complex data objects. For  $0 \leq \sigma \leq 1$ ,  $0 \leq \alpha \leq 1$ , we say that there is a degree  $\sigma$  *knowledge breaching* of  $K$  at the significance level  $\alpha$ , if the adversary is able to define a knowledge service  $f_{K'}$  according to a learned knowledge  $K'$ , so that  $Pr(f_{K'}(\cdot) = f_K(\cdot)) > \sigma$  with a probability  $1 - \alpha$ , where  $Pr(f_{K'}(\cdot) = f_K(\cdot))$  is the

probability that the two services give the same response to the same service request.

**2.3.3. Abuse-Accountability** In certain systems (e.g., the systems coordinating government agencies' counter-terror activities), abuse of knowledge could result in more catastrophic consequences than abuse of data. So we need technical means to hold the abusers accountable. In particular, we must deal with knowledge abusers that typically are corrupt authorized users or insiders. The accountability mechanism should at least help system management identify suspicious activities and even trace them back to the abusers.

As an analogy, traditional information systems have adopted the concept of *auditing*. We stress that abuse-accountability is a much more broad and general concept, because it may rely on existing mechanisms such as auditing and intrusion detections. Moreover, it is highly desirable that the accountability mechanism can be automatically triggered by the transaction data, even if the data are encrypted.

### 3. Related Work

**On the evolution of service oriented 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" [11], "data mining model as a service" [14], and the general notion of "web service". Our PBKM framework emphasizes on security issues of those services that are based on extracted knowledge models. As a specific instantiation of the PBKM, we explored in [16] the notion of "knowledge as a service", where a service provider can be compensated. As another example, consider the following scenario: a life insurance provider may minimize the risk of loss by determining the premium of a new client based on the likelihood of the new client being involved in a fatal car accident, which is a knowledge that a car insurance company could provide. In this context, we investigate knowledge breaching by exploring two specific adversarial strategies: one is based on a new algorithm, and the other is based on a known active machine learning algorithm. Through systematic experiments (with various heuristic optimizations), we showed that knowledge breaching is seemingly inevitable.

**On the relationship to privacy protection.** 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. [2, 3, 4] and their follow-ons). These techniques target at protecting users' anonymity while allowing them to authenticate themselves. 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 [7] 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 [15]. Moreover, a systematic study of the information-disclosure problem in data exchange applications was presented in [13]. We remark that all these techniques do not address the problem of knowledge breaching in the context of knowledge as a service, although inference can be seen as an attack similar to knowledge breaching.

**On the relationship to data mining and machine learning.** On one hand, a PBKMS relies on privacy-preserving knowledge extraction (e.g., data mining) techniques to extract knowledge from raw data, so that the goal of preserving the secrecy of individual data records is achieved while useful patterns are derived. There are two approaches to privacy-preserving data mining. The first approach is to randomize the values in individual records [1] and then to extract a knowledge model from the randomized data, after first compensating for the randomization (at an aggregate level). This approach is potentially vulnerable to privacy breaches since based on the distribution of the data, one may be able to predict 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 [6] for the subtleties in capturing the right notion of privacy) and does not extract accurate knowledge. The second approach is based on cryptographic secure multi-party computation techniques [17, 10, 12]. This approach does extract accurate knowledge and provide a strict privacy guarantee, but is typically much less efficient. In spite of some recent advances in cryptography (e.g., [8]), 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. As mentioned before, in [16] we investigated a knowledge breaching strategy based on an active machine learning algorithm.

### 4. 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, breaching-aware knowledge dissemination, and abuse-accountability. In the following, we outline some of them.

One challenge is to develop practical and provably-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 3, both the perturbation based and the cryptographic privacy-preserving data mining techniques require a lot more further research.

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 mentioned before, what constitutes a knowledge breaching depends on the types of the knowledge service and of the underlying knowledge. However, even for the simple type of knowledge breaching studied in [16] 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, there are many directions for further investigation. We list some of them in the following.

- 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).
- How should an appropriate pricing mechanism (such as a breaching-aware knowledge dissemination policy) be devised? Although a heuristic mechanism can be based on the behavior of a specific breaching strategy, an optimal mechanism would be based on the minimum number of queries required to derive the targeted knowledge. But which breaching strategy is optimal?
- 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).

With regard to abuse accountability, we need to somehow “extract” useful information from large volumes of transaction data, so that the system management can hold abusers accountable. This should be true even if the transaction data is encrypted, and the system management is simply not allowed to require the decryption of the whole data.

Other issues of the knowledge management include efficient and effective storage and retrieval of knowledge, visualization and analysis of knowledge, etc.

## 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] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24:84–88, Feb. 1981.
- [3] 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.
- [4] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, Oct. 1985.
- [5] Data Mining Group. PMML version 2.1. <http://www.dmg.org>, March 2003.
- [6] 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.
- [7] C. Farkas and S. Jajodia. The inference problem: A survey. *SIGKDD Explorations*, 4(2):6–11, 2003.
- [8] 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.
- [9] O. Goldreich. *The Foundations of Cryptography*, volume 2. Cambridge University Press, 2004.
- [10] 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.
- [11] 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.
- [12] 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*.
- [13] 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.
- [14] S. Sarawagi and S. H. Nagaralu. Data mining models as services on the Internet. *ACM SIGKDD Explorations*, 2(1):24–28, 2000.
- [15] 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.
- [16] S. Xu and W. Zhang. Knowledge as a service and knowledge breaching. submitted for publication, 2004.
- [17] A. C. Yao. How to generate and exchange secrets. In *FOCS86*, pages 162–167, Toronto, 1986. IEEE.