Privacy Preserving Data Integration and Sharing

Chris Clifton AnHai Doan Ahmed Elmagarmid
Murat Kantarcioglu Gunther Schadow Dan Suciu
Jaideep Vaidya

The goal of this paper is to identify potential research directions and challenges that need to be addressed to perform privacy preserving data integration. Increasing privacy and security consciousness has lead to increased research (and development) of methods that compute useful information in a secure fashion. Data integration and sharing have been a long standing challenge for the database community. This need has become critical in numerous contexts, including integrating data on the Web and at enterprises, building e-commerce market places, sharing data for scientific research, data exchange at government agencies, monitoring health crises, and improving homeland security.

Unfortunately, *data integration and sharing are hampered by legitimate and widespread privacy concerns.* Companies could exchange information to boost productivity, but are prevented by fear of being exploited by competitors or antitrust concerns. Sharing healthcare data could improve scientific research, but the cost of obtaining consent to use individually identifiable information can be prohibitive. Sharing healthcare and consumer data enables early detection of disease outbreak[16], but without provable privacy protection it is difficult to extend these surveillance measures nationally or internationally. Fire departments could share regulatory and defense plans to enhance their ability to fight terrorism and provide community defense, but fear loss of privacy could lead to liability. The continued exponential growth of distributed personal data could further fuel data integration and sharing applications, but may also be stymied by a privacy backlash. *It is critical to develop techniques to enable the integration and sharing of data without losing privacy.*

The need of the hour is to develop solutions that enable widespread integration and sharing of data, especially in domains of national priorities, while allowing easy and effective privacy control by users. A comprehensive framework that handles the fundamental problems underlying privacy-preserving data integration and sharing is necessary. The framework should be validated by applying it to several important domains and evaluating the result.

Concurrently, various privacy preserving distributed data mining methods have also been developed which mine global data while protecting the privacy/security of the underlying data sites. However, all of these methods also assume that data integration (including record linkage) has already been done. Note that while data integration is related to privacy-preserving data mining, it is still significantly different. Privacy-preserving data mining deals with gaining knowledge *after integration problems are solved.* First, a framework and methods for performing such integration is required.

1 Motivation

There are numerous real-world applications which require data integration while meeting specific privacy constraints. We now discuss some of these "motivating drivers".

1. Sharing Scientific Research Data: Analyzing the prevalence, incidence, and risk factors of diseases is crucial to understanding and treating them. Such analyses have significant impact on policy decisions. An obvious pre-requisite to (carrying out) such studies is to have the requisite data available. First, data needs to be collected from disparate health care providers and integrated while sanitizing privacy-sensitive information.

This process is extremely time consuming and labor intensive. Privacy concerns are a major impediment to streamlining these efforts. A breach of privacy can lead to significant damage (harm) to individuals both materially and/or emotionally. Another problem is the possibility of discrimination against various sub-groups from seemingly conclusive statistical results. Similarly, health care providers themselves risk loss by leaking accurate data reflecting their performance and weaknesses.

Privacy is addressed today by preventing dissemination rather than integrating privacy constraints into the data sharing process. Privacy-preserving integration and sharing of research data in health sciences has become crucial to enabling scientific discovery.

2. Effective Public Safety and Health Care: Integration and sharing between public agencies, and public and private organizations, can have a strong positive impact on public safety. But concerns over the privacy implications of such private/public sector sharing [15] have impacted areas of national priority, including homeland security: The Terrorism Information Awareness program was killed over privacy concerns [10].

Detecting and containing disease outbreaks early is key to preventing life-threatening infectious diseases. Outbreaks of infectious diseases such as West Nile, SARS, and bird flu; as well as threats of bio-terrorism; have made disease surveillance into a national priority. Outbreak detection works best when a variety of data sources (human health-care, animal health, consumer data) are integrated and evaluated in real time.

For example, the Real-Time Outbreak Detection System [16] (at the University of Pittsburgh Medical Center) uses data collected from regional healthcare providers and purchase records of over-the-counter drugs to determine outbreak patterns. This system forwards all regional data to a central data warehouse for evaluation purposes. Although data is de-identified in accordance with HIPAA safe-harbor rules (by removing 19 kinds of identifiers), privacy concerns remain about both patient privacy and organizational privacy (e.g., some participant organizations wish to keep the number of visits by ZIP code secret.)

Public health is largely exempt from U.S. privacy rules, raising the specter of systems with inadequate privacy protection. The concerns are similar to the risks noted above for healthcare research data: External attacks or insider misuse can damage individuals, healthcare providers, or groups within society. Protecting identity and liability exposure by effective privacy-preserving data integration and sharing techniques will enable advances in

emergency preparedness and response, public safety, health care and homeland security that might otherwise be prevented due to privacy concerns.

2 Data Integration and Data Mining

Data Integration and Data Mining are quite closely coupled. Integration is a necessary pre-requisite before mining data collected from multiple sources. At the same time, data mining/machine learning techniques are used to enable automatic data integration. Several systems have been developed to implement automatic schema matching [11, 7, 4]. The systems use machine learning/data mining tools to help automate schema matching. SemInt [11] uses neural networks to determine match candidates. Clustering is done on similar attributes of the input schema. The signatures of the cluster centers are used as training data. Matching is done by feeding attributes from the second schema into the neural network. LSD [7] also uses machine learning techniques for schema matching. LSD consists of several phases. First, mappings for several sources are manually specified. Then source data is extracted (into XML) and training data is created for each base learner. Finally the base learners and the meta-learner are trained. Further steps are carried out to refine the weights learned. The base learners used are a nearest neighbor classification model as well as a Naïve Bayes learner. Again, there has been work on different privacy preserving classification models [17] that is applicable. Artemis [4] is another schema integration tool that computes "affinities" in the range 0 to 1 between attributes. Schema integration is done by clustering attributes based on those affinities. Clearly, a lot of work in both privacy preserving data mining as well as cryptography is relevant to the problem of privacy preserving schema integration. However, it is not yet clear how this could be applied efficiently.

Record linkage also uses various machine learning techniques. Record linkage can be viewed as a pattern classification problem [9]. In pattern classification problems, the goal is to correctly assign patterns to one of a finite number of classes. Similarly, the goal of the record linkage problem is to determine the matching status of a pair of records brought together for comparison. Machine learning methods, such as decision tree induction, neural networks, instance-based learning, clustering, are widely used for pattern classification. Given a set of patterns, a machine learning method builds a decision model that can be used to predict the class of each unclassified pattern. Again, prior privacy preserving work is relevant. At the other end of the spectrum, privacy preserving data mining assumes that data integration has already been done, which is clearly not a solved problem.

3 Privacy Preservation Challenges

As part of the overall problem, we see the following fundamental challenges in privacy-preserving data integration and sharing:

3.1 Privacy Framework

How can we develop a privacy framework for data integration that is flexible and clear to the end users? This demands understandable and provably consistent definitions for building a privacy policy, as well as standards and mechanisms for enforcement.

Database security has generally focused on access control: Users are explicitly (or perhaps implicitly) allowed certain types of access to a data item. This includes work in multilevel secure database as well as statistical queries[1].

Privacy is a more complex concept. Most privacy laws balance benefit vs. risk[8]: access is allowed when there is adequate benefit resulting from access. An example is the European Community directive on data protection which allows processing of private data in situations where specific conditions are met. The Health Insurance Portability and Accountability Act in the U.S. specifies similar conditions for use of data. Individual organizations may define their own policies to address their customers' needs. The problems are exacerbated in a federated environment. The task of data integration itself poses risks, as revealing even the presence of data items at a site may violate privacy.

Some of the privacy issues have been addressed for the case of a single database management system in Hippocratic Databases [3]. Other privacy issues have been addressed for the case of a single interaction between a user and a Website in the P3P standard [6]. None of the current techniques address privacy concerns when data is exchanged between multiple organizations, and transformed and integrated with other data sources.

A framework is required for defining private data and privacy policies in the context of data integration and sharing. The notion of Privacy Views, Privacy Policies, and Purpose Statements is essential towards such a framework. We illustrate using the "Sharing Scientific Research Data" example of Section 1.

Privacy Views The database administrator defines what is private data by specifying a set of *privacy views*, in a declarative language extending SQL. Each privacy view specifies a set of *private attributes* and an owner. By definition, data that appears in some privacy view is considered private; otherwise it is not private. A simple example of a privacy view is given below:

```
PRIVACY-VIEW patientAddressDob
OWNER Patient.pid
SELECT Patient.address, Patient.dob
FROM Patient
```

This privacy view specifies that a patient's `address` and `dob` (date-of-birth) are considered private data *when occurring together*. Similar definitions are possible for fields that specify "individually identifiable information": Sets of attributes that can be used to tie a tuple or a set of tuples in a data source to a specific real-world entity (e.g., a person). Alternatively, administrators may choose to define *database IDs* or *tuple IDs* as private data, both of which could be used to breach privacy over time. In general, privacy views can be much more complex (i.e. by specifying associations between attributes from different tables).

Privacy views could be implemented by a *privacy monitor* that checks every data item

being retrieved from the database and detects if it contains items that have been defined as private. There are two approaches: compile-time (based on query containment) and run-time (based on materializing the privacy views and building indices on the private attributes). Both approaches need to be investigated and tradeoffs evaluated.

Privacy Policies Along with privacy views, it is necessary to have a notion of privacy policies. The database administrator can decide which policy applies to each view. For example, the following two privacy policies could be specified:

```
PRIVACY-POLICY individualData
ALLOW-ACCESS-TO y
FROM Consent x, patientAddressDob y
WHERE x.pid = y.owner and x.type = 'yes'
BENEFICIARY *
```

```
PRIVACY-POLICY defaultPolicy
ALLOW-ACCESS-TO x
FROM patientName x
BENEFICIARY x.owner
```

The first privacy policy states that private data patientAddressDob (defined above) can be released if the owner has given explicit consent, as registered in a Consent table. The second is a default policy which allows access to patient names as long as benefit accrues to the patient. As with privacy views, more complex privacy policies are also possible.

Privacy policies can be enforced by the server holding the data: data items will be shared only if the *purpose statement* of the requester (see below) satisfies the policy. But, in addition, every data item leaving the server should be annotated with *privacy metadata* expressing the privacy policies that have to be applied. These annotations travel with the data, and are preserved and perhaps modified when the data is integrated with data from other sources or transformed.

Query execution becomes much harder, since all privacy views and policies must result in a single piece of privacy metadata; it is not obvious how to do that. Prior work [13] addresses a similar but not identical challenge: how a set of access control policies result in a single, multiple encrypted data instance.

Purpose Statements Finally, once data has been shared and integrated, it eventually reaches an application that uses it. Here, the privacy metadata needs to be compared with the application's stated purpose. A flexible language is required in which applications can state the purpose of their action, and explicitly mention the beneficiary.

3.2 Schema Matching

To share data, sources must first establish semantic correspondences between schemas. However, all current schema matching solutions assume sources can *freely* share their data and schema. *How can we develop schema matching solutions that do not expose the source data and schemas?* Once two data sources S and T have adopted their privacy policies, as outlined in Section 3.1, they can start the process of data sharing. As the first step, the sources must cooperate to create *semantic mappings* among their schemas, to enable the exchange of queries and data [14]. Such semantic mappings can be specified as SQL queries. For

example, suppose S and T are data sources that list houses for sale, then a mapping for attribute list-price of source T is:

```
list-price  =  SELECT price * (1 + agent-fee-rate)
               FROM HOUSES, AGENTS
               WHERE (HOUSES.agent_id = AGENTS.id)
```

which specifies how to obtain data values for list_price from the tables HOUSES and AGENTS of source S.

Creating mappings typically proceeds in two steps: finding matches, and elaborating matches into semantic mappings[14]. In the first step, *matches* are found which specify how an attribute of one schema corresponds to an attribute or set of attributes in the other schema. Examples of match include "address = location", "name = concat(first_name,last_name)", and "list-price = price * (1 + agent-fee-rate)". Research on schema matching has developed a plethora of automated heuristic or learning-based methods to predict matches [14]. These methods significantly reduce the human effort involved in creating matches.

In the second step, a mapping tool elaborates the matches into semantic mappings. For example, the match "list-price = price * (1 + agent-fee-rate)" will be elaborated into the SQL query described earlier, which is the mapping for list-price. This mapping adds information to the match Typically, humans must verify the predicted matches. Furthermore, recent work [14] has argued that elaborating matches into mappings must also involve human efforts.

Schema matching lies at the heart of virtually all data integration and sharing efforts. Consequently, numerous matching algorithms have been developed [14]. All current existing matching algorithms, however, assume that sources can *freely* share their data and schemas, and hence are unsuitable. To develop matching algorithms that preserve privacy, first the following components need to be developed:

Match Prediction: How to create matches without revealing data at the sources, or even the source schemas. An initial step is to start with learning based schema matching.

In learning-based approaches[11, 7], one or more classifiers (e.g., decision tree, Naive Bayes, SVM, etc.) are constructed at source S, using the data instances and schema of S, then sent over to source T. The classifiers are then used to classify the data instances and schema of T. Similarly, classifiers can be constructed at source T and sent over to classify the data instances and schema of S. The classification results are used to construct a matrix that contain a similarity value for any attribute s of S and t of T. This similarity matrix can then be utilized to find matches between S and T.

Schema matching in this approach reduces to a series of classification problems that involve the data and schemas of the two input sources. As such, it is possible to leverage work in privacy-preserving distributed data mining, which have studied how to train and apply classifiers across disparate datasets without revealing sensitive information at the datasets[12].

Human Verification of Matches: Suppose a match m has been found. Now humans at both or one of the sources S and T must examine m to verify its correctness. The goal

is then to make certain such verification is privacy-preserving. The goal is to give humans enough information to verify matches, while preserving privacy. One way to achieve this can be randomly selecting some values for particular attributes and show the user only these values. It can be argued that revealing only few attribute values does not reveal anything useful about the distribution. Since two attributes are found to be similar, it can be argued that few samples does not reveal too much useful information. Definitely, a measure for privacy loss is needed in this context. We will give more details about this in section 3.5.

Mapping Creation: Once a match has been verified and appears to be correct, humans can proceed to the step of working in conjunction with a mapping tool to refine the match into a mapping. In this step, humans typically are shown examples of data, as generated by various mapping choices, and asked to select the correct example. It is necessary to ensure that people are shown data that allows generating mappings, but does not violate privacy.

3.3 Object Matching and Consolidation

Data received from multiple sources may contain duplicates that need to be removed. In many cases it is important to be able to consolidate information about entities (e.g., to construct more comprehensive sets of scientific data). *How can we match entities and consolidate information about them across sources, without revealing the origin of the sources or the real-world origin of the entities?* Record Linkage is the identification of records that refer to the same real-world entity. This is a key challenge to enabling data integration from heterogeneous data sources. What makes record linkage a problem in its own right, (i.e., different from the duplicate elimination problem), is the fact that real-world data is "dirty". In other words, if data were accurate, record linkage would be similar to duplicate elimination. Unfortunately, in real-world data, duplicate records may have different values in one or more fields (e.g. misspelling causes multiple records for the same person).

Record linkage techniques can be used to disclose data confidentiality. In particular, a privacy-aware corporation will use anonymization techniques to protect its own data before sharing it with other businesses. A data intruder tries to identify as many concealed records as possible using an external database (many external databases are now publicly-available). Therefore, anonymization techniques should also be aware of the record linkage techniques to preserve the privacy of the data.

On the other hand, businesses need to integrate their databases to perform data mining and analysis procedures. Such data integration requires privacy-preserving record linkage, that is record linkage in presence of a privacy framework that ensures the data confidentiality of each business. Thus, we need solutions for the following problems:

- Privacy preserving record linkage: that is discovering the records that represent the same real world entity from two integrated databases each of which is protected (encrypted or anonymized). In other words, records are matched without having their identity revealed.

- Record linkage aware data protection: that is protecting the data, before sharing, using anonymization techniques that are aware of the possible use of record linkage, with public available data, to reveal the identity of the records.

- Online record linkage: linking records that arrive continuously in a stream. Real-time systems and sensor networks are two examples of applications that need online data analysis, cleaning, and mining.

3.4 Querying Across Sources

Once semantic correspondences have been established, it is possible to query (e.g., with SQL queries) across the sources. *How do we ensure that query results do not violate privacy policy? How do we query the sources such that only the results are disclosed? How can we prevent the leaking of information from answering a set of queries?* Only a few general techniques exist today for querying datasets while preserving privacy: statistical databases, privacy-preserving join computation, and privacy-preserving top-K queries. In statistical databases, the goal is to allow users to ask aggregate queries over the database while hiding individual data items [1]. Privacy-preserving joins and the more restricted privacy-preserving intersection size computation have been addressed in [5, 2]. Here, each of the two parties learns only the query's answer, and nothing else. The techniques only apply to a specialized class of queries.

Privacy-preserving top-K queries have also recently been studied. Such a query returns just the closest K matches to a query without revealing anything about *why* those matches are close, what the values of the attributes of the close items are, or even which site the closest matches come from. This is accomplished *efficiently* through the use of an untrusted third party: a party that is not allowed to see private values, but is trusted not to collude with any site to violate privacy.

In the applications we envision the data about a single individual is spread across data sources R_i, $i = 1, n$ (vertically partitioned). The data about all individual is expressed as a join $\bowtie_{i=1}^{n} R_i$, and we would like to enable certain queries over this join while preserving privacy. Typically these queries are computed without actually materializing the join. For example if we ask for the cardinality of the join, then it can be computed as $\mid \cap_{i=1}^{n} \Pi_{id}(R_I) \mid$, where id is the join attribute in all relations. This can be done using privacy-preserving intersection algorithms.

Such simple queries only work for cross-sectional counts. Privacy-preserving data mining also provides some building blocks. However, the issue of inference from multiple queries must still be resolved. Issues include categorizing types of queries with respect to privacy policy, ensuring that query processing does not disclose information, and guarding against leakage from a set of queries.

3.5 Quantifying Privacy Disclosure

In real life, with any information disclosure there is *always* some privacy loss. We need reliable metrics for quantifying privacy loss. Instead of simple 0-1 metrics (whether an item is revealed or not), we need to consider probabilistic notions of conditional loss, such as decreasing the range of values an item could have, or increasing the probability of accuracy of an estimate. In general, a starting classification could measure the following: probability of complete disclosure of *all* data, probability of complete disclosure of a *specific* item, probability of complete disclosure of a *random* item. Privacy preserving methods can be evaluated on the basis of their susceptibility to the above metrics. Also some of the existing measures can be used in this direction. For example, one of the popular metrics ($Infer(x \rightarrow y)$) used in database security can be easily applied for measuring privacy loss in schema matching phase. In the original definition $H(y)$ corresponds to entropy of y, and $H_x(y)$ corresponds to conditional entropy of y given x then privacy loss due to revelation of x is given as follows:

$$Infer(x \rightarrow y) = \frac{H(y) - H_x(y)}{H(y)}$$

Note that for schema matching phase,what is revealed to the human for verification can be modeled as revealing x. Although this measure can be used in many different cases, it is hard to calculate the conditional entropies. Therefore, there is need for developing different privacy metrics.

4 Conclusion

In this paper, we presented potential research directions and challenges that need to be addressed in order to achieve privacy preserving data integration. We also pointed out some plausible solution ideas. Though much work remains to be done, we believe that the full potential of privacy preserving data management can only be exploited by achieving privacy preserving data integration. Availability of such tools will also enable us to use distributed data in a privacy preserving way.

References

[1] N. R. Adam and J. C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Comput. Surv.*, 21(4):515–556, Dec. 1989.

[2] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, San Diego, California, June 9-12 2003.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proceedings of the 28th International Conference on Very Large Databases*, pages 143–154, Hong Kong, Aug. 20-23 2002.

[4] S. Castano and V. D. Antonellis. A schema analysis and reconciliation tool environment. In *Proceedings of the Int. Database Engineering and Applications Symposium (IDEAS)*, 1999.

[5] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explorations*, 4(2):28–34, Jan. 2003.

[6] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (P3P1.0) specification, Apr. 16 2002.

[7] A. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of databases: A multistrategy approach. *Machine Learning Journal*, 50:279–301, 2003.

[8] G. T. Duncan, S. A. Keller-McNulty, and S. L. Stokes. Disclosure risk vs. data utility: The r-u confidentiality map. Technical Report 121, National Institute of Statistical Sciences, Dec 2001.

[9] M. Elfeky, V. Verykios, and A. Elmagarmid. TAILOR: A record linkage toolbox. In *Proceedings of the 18th International Conference on Data Engineering*, San Jose, California, Feb. 2002.

[10] M. Lewis. Department of defense appropriations act, 2004, July 17 2003. Title VIII section 8120. Enacted as Public Law 108-87.

[11] W.-S. Li and C. Clifton. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data and Knowledge Engineering*, 33(1):49–84, Apr. 2000.

[12] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3):177–206, 2002.

[13] G. Miklau and D. Suciu. Controlling access to published data using cryptography. In *Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003)*, pages 898–909, Berlin, Germany, Sept. 9-12 2003. Morgan-Kaufmann.

[14] E. Rahm and P. Bernstein. On matching schemas automatically. *VLDB Journal*, 10(4), 2001.

[15] D. Struck. Don't store my data, Japanese tell government. *International Herald Tribune*, page 1, Aug. 24-25 2002.

[16] F.-C. Tsui, J. U. Espino, V. M. Dato, P. H. Gesteland, J. Hutman, and M. M. Wagner. Technical description of RODS: A real-time public health surveillance system. *J Am Med Inform Assoc*, 10(5):399–408, Sept. 2003.

[17] J. Vaidya and C. Clifton. Privacy preserving naïve bayes classifier for vertically partitioned data. In *2004 SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, Apr. 22-24 2004.