PRIVATE-IYE: A Framework for Privacy Preserving Data Integration

Sourav S. Bhowmick[1] Le Gruenwald[2] Mizuho Iwaihara[3] Somchai Chatvichienchai[4]

[1]School of Computer Engineering, Nanyang Technological University, Singapore
[2] School of Computer Science, University of Oklahoma, Norman, USA
[3] Dept of Social Informatics, Kyoto University, Kyoto, Japan
[4] Dept of Info-Media, Siebold University of Nagasaki, Japan

Abstract

Data integration has been a long standing challenge to the database and data mining communities. This need has become critical in numerous contexts, including building e-commerce market places, sharing data from scientific research, and improving homeland security. However, these important activities are hampered by legitimate and widespread concerns of data privacy. It is necessary to develop solutions that enable integration of data, especially in the domains of national priorities, while effective privacy control of the data. In this paper, we present an architecture and key research issues for building such a privacy preserving data integration system called PRIVATE-IYE.

1 Introduction

Data integration, sharing, and mining the integrated data from distributed, heterogeneous, and autonomous sources in order to discover important knowledge have been a long standing challenge for the database and data mining communities. The increasingly exponential growth of distributed personal data could fuel data integration applications to address real life and more importantly *life threatening* problems such as efficient disease control and improved homeland security. However, these important activities have also raised *legitimate and widespread concerns of data privacy* [14]. We illustrate this with the following examples. The first example [11] illustrates how privacy can be compromised in the context of data sharing in health care. The second example illustrates how efficient disease control is hampered due to lack of proper framework for privacy-conscious data sharing.

Example 1 [Clinical data integration] An important indicator for adequate care of diabetes is the participation of affected patients in the following preventive screenings: a blood test of Hemoglobin A1c (HbA1c); a measure of a patient's LDL cholesterol levels; urinalysis; eye exams; and foot exams. Information about patients, disease diagnosis, medications, preventions, and treatment methods is often distributed among heterogeneous databases of different parties such as physicians, pharmacies, laboratories, and health maintenance organizations (HMOs). Although integration of such sources has been addressed by a number of projects, they *do not address the privacy implications* that can be an outcome of the data fusion. For example, consider the Figures 1(a) and 1(b). These tables contain the integrated information about test compliance rates in 2001 [2]. Figure 1(a) contains the mean test compliance rates in a particular county in the United States and their associated standard deviation. Figure 1(b) indicates the general performance of each HMO. Since each HMO considers its own compliance rates for each of these tests (eg. measure of LDL cholesterol) as sensitive data, this information is not displayed. However, *given the aggregate data published by the integrator in both tables, bounds can be inferred about the sensitive values.* For example, suppose HMO1 is snooping for such sensitive data by analyzing these tables. It can use its knowledge of its own compliance rates and the published aggregate data to infer details of other HMOs using a Non-Linear Programming technique (Figures 1(c) and 1(d)). Hence, a data integration system should be able to detect and limit that type of privacy breach. □

Example 2 [Disease Outbreak Control] The recent SARS (Severe Acute Respiratory Syndrome) pandemic has infected more than 8000 people and caused nearly 800 deaths all over the globe; a staggering 10% mortality rate. Many countries such as Singapore, Hong Kong, and China suffered immensely from financial fall-out caused by travel restrictions, restrictions on export and import, etc. Although the importance of discovering effective drugs for SARS is undeniable, such epidemic must be fought by identifying trends and patterns in the disease outbreak, such as understanding and predicting the progression of the disease. This will act as a catalyst for disease control and facilitate drug discovery process. An effective mechanism for achieving this important objective is to use the data warehousing model of gathering all relevant data from different sources to a central repository and then run a set of algorithms against this data to detect trends and patterns. This requires integration and sharing of healthcare data from various relevant local and international sources. This step is extremely important as sharing healthcare data facilitates early detection of disease outbreak [39]. However, in reality, the cost of obtaining consent to use individually identifiable information can be prohibitive as without provable privacy protection it is almost impossible to extend these surveillance measures nationally or internationally. Indeed, according to many experts, the lack of sharing of information by the authorities in China, where SARS was first reported, with other countries may have led to this global health crisis[1]. □

Observe that data encryption techniques and classical role-based access control mechanism are not sufficient to preserve privacy of data in the above examples. This is because state-of-the-art security mechanisms primarily prevent *unauthorized* users to access sensitive data. However, they cannot prevent *authorized* users to do secondary analysis on *authorized* data. Furthermore, access control mechanism governs who can access what objects. Once access is granted, the involvement of the access control ends. Hence, access control mechanism can prevent unauthorized users to view sensitive information about individual tests in Example 1. In other words, authorized users can view only aggregate data as in Figures 1(a) and (b). However, it cannot prevent secondary analysis of such data. That is, it cannot prevent the snoopy HMO1 to infer sensitive information from the aggregate data. Consequently, privacy is a more complex concept compared to data security. Most privacy laws balance benefit against risk [23]; access is allowed when there is adequate benefit resulting from access.

The above examples illustrate that the role of individual privacy rights in the context of critical problems such as national security and disease outbreak is a sensitive issue. Disallowing access to relevant information *completely* because of privacy concern may often be a stumble block for greater beneficial purposes (as shown in Example 2). On the other hand, allowing access to personal information to facilitate critical activities such as tracking terrorists, fraud detection, disease control etc. can result in privacy backlash due to violation of individual privacy rights. In 1986, the complexity and sensitivity of this dual issue was interestingly summarized by the then-Prime Minister of Singapore and founder of modern Singapore Mr Lee Kwan Yew [1]:

> "*I am often accused of interfering in the private lives of citizens. Yet, if I did not, had I not done that, we wouldn't be here today. And I say without the slightest remorse, that we wouldn't be here, we would not have made economic progress, if we had not intervened on very personal matters - who your neighbor is, how you live, the noise you make, how you spit, or what language you use. We decide what is right, never mind what the people think. That's another problem.*"

We believe that there is a need for a comprehensive framework that can catalyze effective functioning of the two conflicting challenging issues: sharing of critical information for greater good while minimizing privacy backlash. Such framework should not only allow data integration, sharing and mining from heterogeneous sources but also preserve privacy of both *sources* and *users*. Note that in the context of data integration there are two entities whose privacy is important to preserve: the *source* or organization (e.g. hospitals) that stores and shares relevant data and *users* (e.g. patients) whose data are stored in the remote sources. While not necessarily an individual privacy issue, protecting the data source may be a prerequisite for organizations to participate in sharing.

In this paper we present a framework for building a deployable system called PRIVATE-IYE (***PRI****vacy Preser****V****ing* ***DA****ta In****TE****grat****I****on* ***SY****st****E****m*) and some of the major research issues associated with this framework. As our framework is still in its formative stages, *our aim of this paper is to present essential research issues in designing the proposed framework*, rather than providing a solution to a specific problem. Our objective is to present the *vision* of our framework, with the hopes that others in the community will explore further on specific problems in the arena of privacy preserving data integration. Note that the primary norm of privacy protection is written laws and regulations (e.g., HIPAA in United States). Our aim, however, is to design a framework that protects privacy beyond today's written law.

The rest of the paper is organized as follows. In Section 2, we present the related works and compare the novelty of our framework. In Sections 3-5, we present the architecture of PRIVATE-IYE and identify the key research issues associated with the various components of the architecture. The last section concludes this paper.

2 Related Research and Novelty

Our proposed privacy preserving data integration system is largely influenced by several technologies as follows.

Data Integration: The key difference between existing works in data integration [22, 26] and ours is that our approach is build on the *foundation of privacy preservation*. This has several significant impacts on the conceptual and algorithmic frameworks of the system. First, all

[1] http://www.computer.org/computer/homepage/1003/profession/

Test	Average Compliance among HMOs	Standard deviation
HbA1c check	83%	5.7%
Lipid profile	54%	4.7%
Eye exam	45%	2.0%

(a) Test Compliance (1)

HMO	Average Performance
HMO1	58%
HMO2	65%
HMO3	60%
HMO4	60%

(b) Test Compliance (2)

	HbA1c	Lipid Profile	Eye Exam	Avg.
HMO1	75.0%	56.0%	43.0%	58.0%
HMO2	?			65.0%
HMO3				60.0%
HMO4				60.3%
Avg.	83.0%	54.1%	45.4%	60.8%
Sigma	5.7%	4.7%	2.0%	

(c) Information known to HMO_1

	HbA1c	Lipid Profile	Eye Exam	Avg.
HMO1	75.0%	56.0%	43.0%	58.0%
HMO2	[87.2; 88.5]	[58.6; 59.8]	[46.8; 47.9]	65.0%
HMO3	[82.8; 86.4]	[48.1; 52.3]	[44.5; 47.2]	60.0%
HMO4	[82.9; 86.7]	[48.6; 53.1]	[44.5; 47.4]	60.3%
Avg.	83.0%	54.1%	45.4%	60.8%
Sigma	5.7%	4.7%	2.0%	

(d) Interval inferred by snooping HMO_1

Figure 1. An example of privacy violation.

current schema matching techniques assume sources can freely share their data and schema. With privacy being the core issue in our integration process, *the issue of developing schema matching solutions that do not expose the source data and schema is of prime importance*. Second, data received from multiple sources may contain duplicates that need to be removed. In privacy centric data integration, discovering records that represent the same real world entity from two integrated databases, each of which is protected, is a challenging problem. Third, it is imperative to ensure that query results do not violate privacy policy. Several techniques have been proposed to preserve privacy for certain types of single query [4] and [8]; unfortunately, *ensuring that a sequence of query results cannot be combined to disclose individual data is still an unsolved problem* [14]. Finally, in real life, with any information disclosure there is always some privacy loss. Hence, we need reliable metrics for *quantifying such privacy loss*.

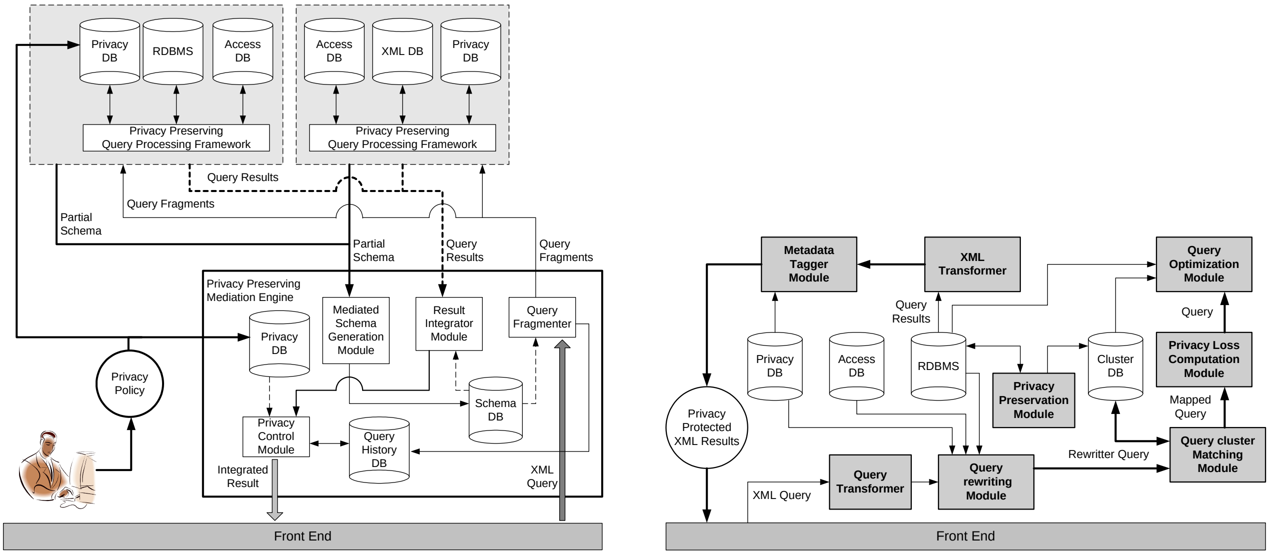
Privacy Preserving Databases: Recently, there have been increasing research efforts by the database community to make privacy as a central concern for databases. An example of such efforts is the *Hippocratic databases* [6]. Another effort is the Platform for Privacy Preferences (P3P), developed by W3C. It provides a way for a web site to encode its data use practices in a machine-readable XML format, known as a P3P policy [17], which can be programmatically compared against a user's privacy preferences expressed in APPEL (XML format) [16]. In [7], Agrawal et al. proposed a server-centric architecture for implementing P3P, in which the P3P policies (in XML format) are shredded into a relational database. Then the privacy preferences in APPEL are transformed to SQL and executed against the database to match the preferences against the privacy policies. None of these efforts addresses privacy concerns *when data is exchanged between multiple organizations, and transformed and integrated with other data sources*.

Secured Databases: The work in this area can be broadly classified into role-based access control and multi-level security. An access control ensures that all direct accesses to the system are authorized according to access rules given by the security policies. The access control governs who can access what objects and ends once access is granted. In role-based access control, access to the data are allowed or prohibited based on the role in which an entity is acting [35]. Multi-level security allows multiple levels of security to be defined and associated with individual attribute values. The security level of a query may be higher or lower than that of individual data items. A query with a lower level of security cannot read a data item requiring higher level of clearance, while a higher security query cannot write a lower security data item. Two queries having different levels of security can, thus, generate different answers over the same database.

All the above approaches give access to data objects *without considering the privacy preferences* of these data objects. This is because these approaches primarily prevent *unauthorized* users from accessing sensitive data. However, they cannot prevent *authorized* users from doing secondary analysis on *authorized* data (as shown in Example 1). It is also possible for these techniques to violate privacy without performing any secondary analysis over data.

Statistical Databases: Research in this area has focused on enabling queries on aggregate information (e.g. sum, count) from a database without revealing individual records [4]. It can be broadly classified into data perturbation and query restriction. Data perturbation involves either altering the input database [32, 38], or altering query results returned [10, 20]. Query restriction includes schemes that check for possible privacy breaches by keeping audit trails [13] and controlling overlap [21] of successive aggregate queries. The techniques developed have focused only on aggregate queries and relational data types. New mechanisms will be needed to perturb the richer information types and audit the richer queries in our framework.

Privacy Preserving Mining: Classical data mining systems operate by gathering all data into a central site. However, privacy concerns can prevent building a centralized warehouse. This has led to a growing body of research effort on the development of *privacy preserving data mining* algorithms. These algorithms can be divided into two different groups. One approach adopts a distributed framework [33, 18, 30]; the other approach adds random noise to the data in such a way that the individual data values are distorted while still preserving the underlying distribution



(a) Privacy preserving query processing.

(b) Privacy preserving mediation engine.

Figure 2. PRIVATE-IYE **architecture.**

properties [5, 29]. These techniques assume that data has been integrated prior to their applications. *However, in reality, sources may not be willing to give their data in the first place unless privacy of the data is preserved [14].* In this research, we address the problem of facilitating such privacy preserving integration. While we expect some of the data perturbation techniques to find applications in our framework, it is clear that they are not foolproof in protecting data privacy [29]. Hence, we need a safer and more efficient method for data perturbation.

3 Privacy Policy Formulation Framework

The architecture of our system is shown in Figure 2(a). It is similar in spirit to the architecture of research prototypes [25], with the distinguishing factor that our system is built upon the privacy preservation theme using an XML data model. XML provides much greater flexibility in the kinds of data that can be handled by our system. In addition to the common case of relational data, we can naturally handle data from hierarchical stores and data in structured files. Our architecture consists of three major components: the *privacy policy formulation framework*, the *privacy preserving query processing* framework, and the *privacy preserving mediation engine*. In this section, we begin with the first framework. The objective of the *privacy policy formulation framework* is to provide a mechanism for defining private data and privacy policies for both sources and users in the context of data integration and sharing. Although there are works done in defining access rules, not much systematic research has been done in defining privacy policies.

The source or user specifies its privacy policies and views using a user-friendly interface that are stored in the remote source as well as in the Mediator Engine. The justification of storing privacy policies in the mediator is as follows. The preservation of privacy in PRIVATE-IYE is achieved at two levels. First, at each remote source, a privacy conscious query evaluation framework is created to ensure that the results coming out from a specific source do not violate the privacy of the data/source. We shall discuss this further in Section 4. Second, the mediator engine further verifies that the *integrated* query results from all remote sources do not violate the privacy (even if the query results from a specific source may not violate the privacy). In order to do this the mediator must have knowledge about the privacy policies that are relevant to the query results. Hence, the policies are stored in the mediator engine as well.

In order to realize this framework, the following three flexible declarative languages need to be developed. The first language enables users to specify how different personal data items can be shared under a specific stated purpose by the requester and in a specific form (exact value, aggregate, range, etc.). Note that the objective is to design a language that can be seamlessly used by users to specify private data and control the level of legitimate information/privacy loss they can allow to happen. The second language defines private data in a source by specifying a set of *privacy views*. And the third language defines privacy policies of remote sources. Data items in a source can be shared only if the purpose statement of the requester satisfies the policy. Recently, there is a systematic effort towards privacy-enhanced authorization model and language for defining and enforcing for flexible access restrictions [9].

In this context, we can either extend SQL or extend XQuery (XML query language) to develop the above languages. As relational databases are prevalent, using an extended SQL can simplify policy enforcement. However, this becomes a cumbersome solution for sources containing hierarchical data or structured files. In this situation, an extended XQuery shall be more advantageous. Moreover, as an XML-based query language is used in PRIVATE-IYE to pose queries on sources, it is much easier to integrate it with XML-based privacy control policies.

4 Privacy Preserving Query Processing Framework

This component is built at a remote source to provide a framework that can preserve privacy and access constraints while evaluating requesters' queries. It minimizes privacy loss when query results are returned to the requesters/applications. Traditional data integration systems do not create such a framework at the source as they assume that data is *freely* accessible. However, in a privacy preserving data integration system, such a framework is significant as it is necessary to preserve privacy of data and source *before* it can be transferred or exchanged with other sources/applications.

Figure 2(a) shows the architecture of the privacy preserving query processing mechanism at a remote source. This architecture can easily be extended to a set of remote sources. The *Query Transformer* module transforms the fragmented XML query from the *Mediation Engine* into an appropriate query language for the destination source. For example, if an RDBS is being queried, then it generates SQL. We do not perform this transformation in the *Mediation Engine* because unlike traditional data integration systems, due to privacy constraints, the schemas of some of the underlying sources may not be available to the Engine in order to perform the transformation. Furthermore, the XML query may be "*approximately*" formulated as the mediator may not provide enough information to guide the requester to formulate exact queries.

The *Query Rewriter Module* takes as input the query that needs to be processed on the site and *rewrites* it. This module essentially examines the authorization rules (stored in *AccessDB*), privacy policies and preferences (stored in *PrivacyDB*), and metadata corresponding to the requested data, and produces a query that will only retrieve the information that can be accessed by the requester as well as preserves the privacy of the data. The modified query is then sent to the *Query Cluster Matching Module* to identify appropriate privacy preserving techniques to use on the query results.

The query rewriting mechanism may protect the privacy of certain data items. However, it may not prevent the query results to be analyzed or mined further by the requester to extract sensitive information (as in Example 1). To minimize such a privacy violation, it is necessary to understand the characteristics of the query results and possible privacy violations against the results in order to determine the privacy preservation techniques that can be applied on the query results to minimize or prevent such a violation. The *Privacy Preservation Module* is responsible for inferring possible types of privacy breaches for different classes of queries by mining the raw data. It also stores different types of privacy preservation techniques that need to be applied to the data to address these breaches.

Given a rewritten query as input, the goal of the *Query Cluster Matching Module* is to determine the types of privacy preservation techniques that should be applied on the query results. This can be realized by any of the following two ways: (1) execute the query in the database and analyze the query results to determine the privacy preservation techniques that need to be applied (note that in this case, the results of every query posed by the requester needs to be analyzed), and (2) analyze only the *features* of the query (e.g., types of predicates, execution of related queries in the history, types of data returned, etc.) to determine the characteristics of the query results (without executing the query) and corresponding privacy breaches. Then, determine the privacy preservation techniques that need to be applied on the query results. We choose the later approach for the following reasons. First, it is possible to cluster a set of queries having similar privacy breaches by analyzing the query characteristics. Then, similar privacy preservation techniques can be applied to this set of queries. Given a query q, we first map it to a cluster C such that q has similar privacy breaches to the queries in C as they share similar characteristics. Then we can identify the type of privacy preservation techniques that needs to be applied to the query results of q by analyzing the *cluster features* of C stored in *ClusterDB*. The *ClusterDB* contains a set of clusters where each cluster represents a set of queries having similar privacy breaches and, hence, similar privacy preservation techniques. Second, by deferring the execution of the query, we can exploit the features of privacy preservation techniques determined during this process to design efficient query execution plan during query optimization.

Next, the query is analyzed by the *Privacy Loss Computation Module* to quantify the loss of privacy against the loss of information. This information along with that from the previous module are fed to the *Query Optimization Module* which is built on top of a traditional query optimizer to exploit privacy-preservation features while optimizing queries. Finally, the query is sent to the database for execution. Note that the *Privacy Preservation Module* interacts with the query execution process to preserve privacy of the results. Upon successful execution of the query, the *XML Transformer Module* transforms the results into desired XML format. The *Metadata Tagger Module* annotates the results with privacy metadata expressing the privacy policies that have to be applied. Finally, the tagged query results are returned to the *Mediation Engine*. We now present the key research problems that need to be solved in order to realize this framework.

Query transformation (Query Transformer Module): Let q be the XML query fragment that is forwarded by the *Mediation Engine* to the target source S. The XML query is first transformed into the local language (if necessary). This is a non-trivial problem as the query fragment q from the Mediation Engine may be "approximately" constructed as the mediated schema may not contain sufficient information to formulate exact queries. Hence, it is required to be transformed into meaningful query(s) that can be executed against the local database.

Privacy preserving query rewriting (Query Rewriting Module): There are two alternatives for ensuring that the query results do not violate the access rules and data privacy. First, we execute the transformed q on S and then filter out the result instances or attribute instances in the results that violate the access rules and data privacy. Second, we first rewrite the transformed q to q' by *integrating* the relevant privacy policies and access rules with q and then execute q' against S. We choose the latter because by preprocessing the query we shall be able to reduce the cost of execution as it will operate on a smaller set of data in the database. Note that the rewriting algorithm may generate more than one modified query. Hence, it is imperative to design an algorithm that can generate q' such that the privacy loss is minimum.

Privacy preservation techniques (Privacy Preservation Module): We need techniques that can be applied on the query results in order to preserve the privacy of the data and sources. Note that some degree of privacy preservation can be achieved by using a traditional approach of hiding attributes as in classical access control techniques and aggregation of information as in statistical databases. However, these methods are not enough to protect all types of privacy violation. The challenging task is to prevent undesirable analysis and mining of query results. That is, we need techniques to allow people to retrieve individual facts, but we want to protect any generalization and undesirable inference [24] that can be formed from mining the data. In this context, we wish to explore issues such as determining possible privacy breaches for the given query results, data augmentation to preserve privacy, exploit limitations of data mining algorithms to prevent privacy breaches, profiling hard-to-mine data, etc [15]. We also intend to study and extend some of the techniques used in privacy preserving data mining in order to use them in our framework [3, 5, 19, 29, 30, 31, 34, 40, 41].

Privacy-conscious query clustering (Query Cluster Matching Module): There are two major research issues for this problem; *cluster generation* and *query mapping*. For *cluster generation*, we need ways to define and measure *similar* queries based on similarity of query features and privacy breaches. As mentioned earlier, queries with similar privacy breaches have similar privacy preservation techniques. A set of such similar queries Q_s can be considered as a cluster C_i. Ideally, a set of queries in the same cluster has similar privacy breaches to those of the queries in different clusters. That is, if $q_1 \in C_1$ and $q_2 \in C_2$ then q_1 and q_2 result in different types of privacy breaches and, hence, different privacy preservation techniques are applied on the results of q_1 and q_2, respectively. Each cluster is associated with a set of *features* that defines the characteristics of the queries inside the cluster. We need ways to define such features as they are used for *query mapping*. Given a rewritten query q', the goal of *query mapping* is to develop a mechanism such that $C_i = Map(q', C)$ where C is the set of clusters in the *ClusterDB* and C_i is the cluster containing queries that are most similar to q' and, hence, share similar privacy preservation techniques.

Privacy metrics (Privacy Loss Computation Module): We need ways to define and measure privacy so that privacy preserving data integration results do meet actual privacy constraints. We need reliable metrics for quantifying privacy loss. Instead of boolean 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. Also, anonymity is an established measure of privacy, including concepts such as *k-anonymity* [37, 28].

Privacy-conscious query optimization (Query Optimization Module): With the additional costs of privacy checking during query processing and possible results perturbation to preserve privacy, we need novel query processing techniques to reduce these costs. These techniques need to be integrated with the query optimization mechanism so that the most efficient query execution plan incorporates the most efficient privacy checking and preservation plan. Furthermore, the maximum information loss or privacy loss allowed as specified by the requester can also be used in the query plan to filter out irrelevant processing of data.

Privacy preservation for a sequence of queries (Privacy Preservation Module): How can we prevent the leaking of information from answering a set of queries? That is, even if we ensure that the results of a given query do not violate privacy policies and access rules, how do we ensure that a set of query results from a set of queries (these queries may be on the same source or different sources) cannot be combined together to violate data privacy? There have been some works in this direction. For example, [21] gives criteria where a set of queries can be shown to prevent inference of individual values, but this requires tracking queries. A practical way of solving this problem is to establish a *class of queries* that need to be answered, and determine criteria for ensuring that a set of queries from this class provably prevent privacy breaches [14].

5 Privacy Preserving Mediation Engine

Data integration solutions have been largely based on two opposing approaches: warehousing and virtual querying [26]. A cornerstone of our architecture is that our *Mediation Engine* allows us to query on demand (virtual querying) as well as materialize some data locally (warehousing). We take the hybrid approach due to the *quick-response needed during emergency situations* (eg. disease outbreak, bioterrorism). Figure 2(b) shows the architecture of the privacy preserving mediation engine. The *Mediated Schema Generation Module* is responsible for creating a *partial* structural summary of the remote sources (called *mediated schema*). The structural summary acts as a guide for query formulation by the requester. Note that due to privacy concerns, some of the schemas of the remote sources may be unavailable or partially available. When a query is posed using the mediated schema, it is parsed and broken into multiple fragments of *extended* XQueries by the *Query Fragmenter Module*. The fragmented queries are then sent to the target sources and, after execution, the query results from different sources are integrated by the *Result Integrator Module*. The *Privacy Control Module* computes the aggregated privacy loss of the integrated results and verifies whether the integrated results satisfy the privacy constraints of the remote sources. If they do, then the results are returned back to the requester. If some of the results violate privacy constraints, then they are not included in the result set sent to the requester and the remote source(s) is notified about the violation. Our framework allows the integrated results to be warehoused locally for further analysis and mining. We now present the key research problems that need to be solved in order to realize this framework.

Privacy Preserving Schema Matching (Mediated Schema Generation Module): Schema matching lies at the heart of virtually all data integration and sharing efforts. In traditional data integration systems, it is facilitated by creating semantic mappings among the schemas of the sources assuming that the sources are willing to cooperate [36]. However, in our proposed research the schemas of some sources may not be available freely due to privacy constraints. Hence, mapping schemas to generate mediated schemas is a challenging problem. An initial step is to start with learning-based schema matching as highlighted by Clifton et al. [14].

Privacy Preserving Mediated Schema Generation (Mediated Schema Generation Module): The aggregation of (partial) schemas of remote sources is built on the framework of schema matching. We need to develop techniques to generate a meaningful structural summary of the remote sources without violating the privacy constraints of the remote sources.

Design of Privacy-conscious Query Language: In traditional data integration environments, as schema information is freely available, the structural summary of the traditional mediated schema serves as an accurate guide to formulate meaningful queries. However, in PRIVATE-IYE, the mediated schema may not contain sufficient information to enable formulation of semantically accurate queries. Hence, our framework must provide a declarative language that supports *loosely* structured queries. For instance, *date of birth* of a *patient* may be referenced as *dob* in a remote source. As we use an XQuery framework to formulate queries, it can be accessed using the path expression $//patient//dob$. However, the mediated schema may not be aware of the attribute *dob* as the remote source considers this as sensitive information. While formulating queries, the requester may wish to retrieve *dates of birth* of patients but the mediated schema does not provide information about the nominal identifier of this attribute in the remote source. Consequently, if the requester uses the path expression $//patient//dateOfBirth$ in the query then the relevant results will not be retrieved from the remote source. Additionally, the requester should be able to provide the *purpose* of the query and the *maximum information loss* he/she is willing to accommodate in the integrated results. Hence, it is necessary to extend XQuery to support privacy-conscious query formulation.

Query Fragmentation (Query Fragmenter Module): We need ways to define a mechanism of fragmenting the query formulated by the requester in the *Query Fragmenter Module*. This is a challenging problem as the mediated schema may not have sufficient information to accurately fragment the queries and then send to relevant remote sources. Sending queries to irrelevant sources affects adversely the efficiency of the integration process. Hence, it is necessary to design intelligent techniques for query fragmentation and determination of relevant sources with high accuracy.

Results Integration (Result Integrator Module): Unlike results integration in traditional data integration systems, the privacy preservation flavor in our proposed data integration system raises some novel challenges. First, it is necessary to protect the confidentiality of data *after* it has been integrated with data from other sources. As discussed in the preceding section, a privacy-aware source (source having a privacy preserving query processing framework) may use anonymization and data perturbation techniques to protect its own data before sharing it with other sources. A data intruder can still identify as many concealed records as possible by integrating them with external databases (as in Example 1). Hence, it is necessary to devise novel strategies to prevent such privacy breaches. Second, it is necessary to remove duplicates and clean "dirty" data while performing the integration. Although this is also necessary in traditional data integration systems, in a privacy preserving data integration system, such object matchings have to be done without revealing the origins of the sources or the real world origins of the entities.

Computation of Aggregated Privacy Loss (Privacy Control Module): In the preceding section, we have identified the need to explore reliable metrics for quantifying privacy loss in a remote source. This allows the remote source to measure the preservation of privacy against information loss *independent of other sources*. However, the computed value of privacy loss in a source may not hold after the results are integrated with other sources. Suppose a remote source is willing to share the results of a query if the privacy loss is less than k. For a query q, let privacy loss be $k' < k$. It may be possible that after the result is integrated with other sources, $k' > k$. Hence, we need techniques to compute *aggregated privacy loss* of the integrated data from various sources in order to ensure that the integrated results satisfy the privacy constraints of remote sources.

6 Conclusions

Data integration has been a long standing challenge to the database and data mining communities. However, this is hampered by legitimate and widespread concerns of data privacy. The need of the hour is to develop solutions that enable integration of data, especially in the domains of national priorities, while effectively control the privacy of the data. In this paper, we present the architecture and key research issues for building such a privacy preserving data integration system called PRIVATE-IYE. Rather than discussing solutions to a specific problem, we present the *vision* of our privacy preserving data integration framework, with the hopes that others in the community will explore further on specific problems in this arena. We believe that this is an important new application area for data integration, combining commercial interest with intriguing research questions.

Acknowledgement: This material is based upon work supported by (while serving at) the National Science Foundation (NSF). Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] "Lee Kwan Yew's Speech at National Day Rally." *The Straits Times, April 20, 1987*, cited in *Christophen Tremewan*, `http://www.privacyinternational.org/survey/phr2003/countries/singapore.htm`.

[2] PHC4 Pittsburg Healthcare Cost Containment Council (2002). *Diabetes Hospitilization Report 2001* Data, Pittsburg, November 2002. `http://www.phc4.org/adobe/Diab01.pdf`.

[3] D. AGRAWAL AND C. C. AGGARWAL. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In *Proc. of PODS*, 2001.

[4] N. R. ADAMS AND J. C. WORTMAN. Security-control methods for statistical databases: A Comparative Study. *ACM Comuting Surveys*, 21(4), 1989.

[5] R. AGRAWAL AND R. SRIKANTH Privacy-preserving data mining. In Proc. of SIGMOD, 2000.

[6] R. AGRAWAL, J. KIERMAN, R. SRIKANT, Y. XU. Hippocratic Databases. In *Proc. of VLDB* , 2002.

[7] R. AGRAWAL, J. KIERMAN, R. SRIKANT, Y. XU. Implementing P3P Using Database Technology. *In Proc. of ICDE* , 2003.

[8] R. AGRAWAL, A. EVFIMIEVSKI, R. SRIKANT. Information Sharing Across Private Databases. *In Proc. of ACM SIGMOD* , 2003.

[9] C. ARDAGNA, E. DAMIANI, S. DE CAPITANI DI VIMERCATI, P. SAMARATI. Towards Privacy-Enhanced Authorization Policies and Languages. In *Proc. of 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Storrs, USA, 2005.

[10] I. I. BECK. A Security Mechanism for Statistical Databases. *ACM TODS*, 5(3), 1980.

[11] C. BOYENS, R. KRISHNAN, R. PADMAN. On Privacy-Preserving Access to Distributed Heterogeneous Healthcare Information. *In Proc. of HICSS*, 2004.

[12] S. CASTANO, M. G. FUGINI, G. MARTELLA, P. SAMARATI. *Database Security*, Addison-Wesley and ACM Press, 1985.

[13] F. CHIN, G. OZSOYOGLU. Auditing and Inference Control in Statistical Databases. *IEEE TSE*, 8(6), 1982.

[14] C. CLIFTON, M. KANTARCIOGLU, A. DOAN ET AL. Privacy Preserving Data Integration and Sharing. *In Proc. of DMKD* , 2004.

[15] C. CLIFTON, D. MARKS. Security and Privacy Implications of Data Mining. *In Proc. of DMKD*, 1996.

[16] L. CRANOR, M. LANGHEINRICH, M. MARCHIORI. A P3P Preference Exchange Language 1.0 (APPEL1.0). *W3C Working Draft* , 2002.

[17] L. CRANOR, M. LANGHEINRICH, M. MARCHIORI ET AL. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. *W3C Recommendation* , 2002.

[18] W. DU, M. J. ATALLAH. Secure multi-party computation problems and their applications: A review and open problems. In *New Security Paradigms Workshop*, pages 11-20, 2001.

[19] W. Du, Z. Zhan. Using Randomized Response Techniques for Privacy-Preserving Data Mining. *In Proc. of ACM SIGKDD*, 2003.

[20] D. Denning. Secure Statistical Databases with Random Sample Queries. *ACM TODS*, 5(3), 1980.

[21] D. Dobkin, A. Jones, R. Lipton. Secure Databases: Protection Against User Influence. *ACM TODS*, 4(1), 1979.

[22] R. Domenig, K. Dittrich. An Overview and Classification of Mediated Query Systems. *SIGMOD Record*, 28(3), 1999.

[23] G. T. Duncan, S. A. Keller-McNulty, S. L. Stokes. Disclosure Risk vs Data Utiliy: The R-U Confidentiality Map. *National Institute of Statistical Sciences*, Tech Report. 121, Dec 2001.

[24] C. Farkas, S. Jajodia. The Inference Problem: A Survey. *In ACM SIGKDD Explorations Newsletter*, 4(2), 2002.

[25] H. Garcia-Molina, Y. Papakonstantinou et al. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems* 8(2):117–132, 1997.

[26] R. Hull. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. In *Proc. of PODS*, Tucson, 1997.

[27] S. Jajodia, R. Sandhu. Toward a Multilevel Secure Relational Data Model. In *Proc. of the ACM SIGMOD*, 1991.

[28] W. Jiang, C. Clifton. Privacy-Preserving Distributed k-Anonymity. In *Proc. of 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Storrs, USA, 2005.

[29] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the Privacy Preserving Properties of Random Data Perturbation Techniques. In *Proc of ICDM*, 2003.

[30] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In *SIGMOD Workshop on Data Mining and Knowledge Discovery*, 2002.

[31] M. Kantarcioglu, J. Jin, C. Clifton. When do Data Mining Results Violate Privacy? In *Proc. of ACM SIGKDD*, 2004.

[32] C. K. Liew, U. J. Choi, C. J. Liew. A Data Distortion by Probability Distribution. *ACM TODS*, 10(3), 1985.

[33] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Advances in Cryptology*, 2000.

[34] H. Polat, W. Du. Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques. *In Proc. of ICDM*, 2003.

[35] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. A Model of Authorization for Next-generation Database Systems. *ACM TODS*, 16(1), 1991.

[36] E. Rahm, P. Bernstein. On Matching Schemas Automatically. *VLDB Journal*, 10(4), 2001.

[37] P. Samarati, L. Sweeney. Protecting Privacy when Disclosing Information: k-anonymity and its Enforcement Through Generalization and Suppression. *In Proc of IEEE Research in Security and Privacy*, Oakland, 1998.

[38] J. Traub, Y. Yemini, H. Woznaikowski. The Statistical Security of a Statistical Database. *ACM TODS*, 9(4), 1984.

[39] F. -C. Tsui, J. Espino, V. M. Dato, P. Gesteland et al. Technical Description of RODS: A Real-time Public Health Surveillance System. *J Am Med Inform Assoc*, 10(5), pp. 399—308, Sept 2003.

[40] K. Wang, P. S. Yu, S. Chakraborty. Bottom-Up Generalization: A Data Mining Solution to Privacy Protection. *In Proc. of ICDM*, 2004.

[41] N. Zhang, S. Wang, W. Zhao. A New Scheme on Privacy-Preserving Data Classification. *In Proc. of SIGKDD*, 2005.