Technical Report

TR-2009-007

From Droplets to Cloud: Enabling Privacy-Preserving Data Federation

by

Pawel Jurczyk, Li Xiong

Mathematics and Computer Science

EMORY UNIVERSITY

From Droplets to Cloud: Enabling Privacy-Preserving Data Federation

Pawel Jurczyk and Li Xiong
Department of Math&CS
Emory University
Atlanta, GA, USA

Contact information

Pawel Jurczyk, Mathematics & Computer Science, Mail Stop: 1131-002-1AC, Emory University, Atlanta, GA 30322. Tel: (404) 727-5603; E-mail: pjurczy@emory.edu

Li Xiong, Mathematics & Computer Science, Mail Stop: 1131-002-1AC, Emory University, Atlanta, GA 30322. Tel: (404) 727-5603; E-mail: lxiong@emory.edu

Abstract word count: 116
Body word count: 4,579

Abstract

The emergence of cloud computing implies and facilitates managing large collections of highly distributed, autonomous, and possibly private databases. While there is an increasing need for services that allow integration and sharing of various data repositories in the cloud, it remains a challenge to ensure the privacy, interoperability, and scalability for such services. In this paper, we report our research efforts and experiences in developing an infrastructure for scalable and privacy preserving data federations of distributed and possibly private data sources. Our system utilizes a decentralized architecture consisting of individual system nodes, or droplets, which form a virtual server, or cloud, to deliver a seamless and transparent data federation service for users.

1 Introduction

With the trend of cloud computing[1][2], data and computing are moved away from desktop and are instead provided *as a service* from the cloud. Current major components under the cloud computing paradigm include infrastructure-as-a-service (such as EC2 by Amazon), platform-as-a-service (such as Google App Engine), and application or software-as-a-service (such as GMail by Google). There is also an increasing need to provide data-as-a-service [1]

[1] http://en.wikipedia.org/wiki/Cloud_computing
[2] http://www.theregister.co.uk/2009/01/06/year_ahead_clouds/

with a goal of facilitating access to a wealth of data across distributed, heterogeneous and possibly private data sources available in the cloud.

Application scenarios. In the healthcare domain, a national health information technology agenda is to enable the creation of a Nationwide Health Information Network (NHIN)[3], a network of networks that will enable the use of health information for clinical decision making and beyond direct patient care to improve public health. For instance, consider a system that integrates the air and rail transportation networks with demographic databases and patient databases in order to model the large scale spread of infectious diseases (such as the SARS epidemic or pandemic influenza). Rail and air transportation databases are distributed among hundreds of local servers, demographic information is provided by a few global database servers and patient data is provided by groups of cooperating hospitals.

Another example is the Shared Pathology Informatics Network (SPIN)[4] initiative by the National Cancer Institute. The objective is to establish an Internet-based *virtual* interface or service that will allow investigators access to data that describe archived tissue specimens across multiple institutions while still allowing those institutions to maintain local control of the data.

Research challenges. While these data-as-a-service scenarios demonstrate the increasing needs for integrating and querying data across distributed and autonomous data sources, it remains a challenge to ensure privacy, interop-

[3]Nationwide Health Information Network (NHIN). http://www.hhs.gov/healthit/healthnetwork/background/

[4]Shared Pathology Informatics Network. http://www.cancerdiagnosis.nci.nih.gov/spin/

erability, and scalability for such data services. To achieve interoperability and scalability, data federation is increasingly becoming a preferred data integration solution. In contrast to a centralized data warehouse approach, data federation combines data from distributed data sources into one single *virtual* data source, or data service, which can then be accessed, managed and viewed as if it was part of a single system. Indeed, the NHIN will not include a national data store or centralized systems. Instead, it will use shared architecture (services and standards) to interconnect health information exchanges.

In addition, there are two important privacy constraints to be considered for such data federation services. The first constraint is the privacy of individuals or *data subjects* (such as patients). For example, personal health information is protected under the Health Insurance Portability and Accountability Act (HIPAA)[5][6]. This constraint can be addressed by data anonymization or de-identification which transforms the data through techniques such as attribute removal or generalization so that it does not contain individually identifiable information. In fact, the problem of data anonymization for a *single* database (in client-server setting) has been extensively studied in the database community in recent years. However few works have studied the anonymization problem for distributed data federation services.

[5]Health Insurance Portability and Accountability Act (HIPAA). http://www.hhs.gov/ocr/hipaa/.

[6]State law or institutional policy may differ from the HIPAA standard and should be considered as well.

The second constraint is the privacy of *data providers* (e.g. institutions) as they may not want to reveal part of their data or the ownership of the data for competition and various other reasons. For example, a hospital may consider its test compliance rates as sensitive or may not want to reveal its admission of a particular group of patients. This constraint can be potentially addressed by *secure multi-party computation* approaches where we wish to compute an answer given a query spanning multiple databases without revealing any information of each individual database apart from the query result. However, these techniques are insufficient when the query results alone (without additional intermediate information) reveal certain ownership of the data.

While there are extensive works dedicated to each of the above aspects and techniques, very few have taken a systems approach to study them in the context of data federation services for multiple distributed and autonomous data sources. We set out to explore research opportunities directed towards integrating these building blocks through the development of a new privacy-preserving data federation architecture for delivering data services for the cloud.

Contributions. In this article we describe an architecture for building scalable and privacy-preserving data federation services for distributed and possibly private databases in the cloud. We mainly focus on system architecture issues and report our efforts and experiences in building such a system. Our system consists of a few innovative components and contributions.

First, it provides a *secure distributed anonymization* component for constructing a *virtual* anonymized database from multiple data providers while preserving privacy for both *data subjects* and *data providers*. In addition, it provides a *secure distributed query processing* engine for querying the virtual anonymized data in a scalable and privacy-preserving way. Secure query operators are *integrated* into the query processing engine to preserve privacy for data providers in a transparent manner. It is worth noting that some data sources may contain personal data that require anonymization while others may not. When sensitive information is being queried, it needs to be recognized automatically and transparently from users point of view and secure operators will be deployed to guarantee privacy of the data.

Second, it builds on top of a *distributed* mediator-wrapper architecture where individual system nodes serve as mediators (mediating queries across data sources) and/or wrappers (retrieving data from individual data sources). They interact with each other in a P2P fashion and form a *virtual* system to provide a seamless and transparent data federation service. As an analogy, our system nodes can be considered as *droplets*, small elements that provide similar functionality in the cloud. An element can be a single physical machine or a service provided by a physical machine (in that case physical machine can function as several droplets). Just as thousands or millions of droplets form a single drop in nature, in cloud computing, groups of *droplets* that provide similar functionality can form a *micro-cloud*. *Micro-clouds* are an integral part of the whole cloud computing system and can provide spe-

cific services to users. In spirit, our data federation service which we propose here can be considered as such *micro-cloud*.

We realize that there are alternative approaches and many challenging issues in building a full-fledged system. As a result, we focus on the class of applications with horizontal partitioned data like the SPIN scenario where data are not as dynamic. This allows us to move forward in building an integrated system while studying issues such as dynamic data sources at a later stage and deferring certain issues such as schema integration, access control and query auditing to future research agenda. We believe that given the complexities of the problem, by focusing on simpler scenarios and a subset of issues in the beginning, the system and techniques will still have practical importance and the lessons we learn from our experience in building the system will also be valuable.

2 Related Work

Privacy preserving data publishing. The problem of protecting privacy for *data subjects* in the released data for a single database has been extensively studied in recent years. Since the seminal work on k-anonymity [2], a large body of work contributes to data anonymization that transforms a dataset to meet a privacy principle (e.g. k-anonymity) using techniques such as generalization, suppression (removal), permutation and swapping of certain data values so that it does not contain individually identifiable in-

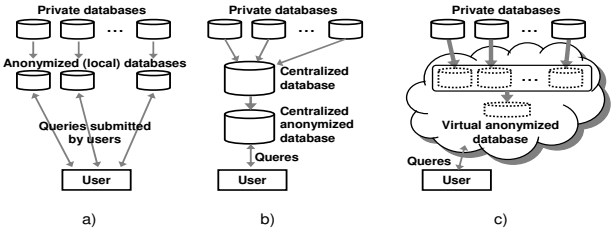

Figure 1: Possible architectures for privacy preserving distributed data publishing.

formation [3]. In general, the goal of k-anonymity protocols is to transform initial data set so that each individual is identical with at least $k-1$ other individuals.

There are a number of potential approaches one may apply to enable privacy preserving data publishing for distributed databases. A naive approach is for each data custodian to perform data anonymization independently as shown in Fig. 1a. One main drawback of this approach is that data is anonymized before the integration and hence will cause the data utility to suffer. In addition, individual databases reveal their ownership of the anonymized data. An alternative approach assumes an existence of third party that can be trusted by each of the data owners as shown in Fig. 1b. However, finding such a trusted third party is not always feasible. Compromise of the server by hackers could lead to a complete privacy loss for all participating parties.

Finally, there is distributed data anonymization approach as illustrated in Fig. 1c. There are some works along this direction for distributed data

providers to construct an integrated and anonymized database. [4] presented a two-party framework that generates k-anonymous data from two vertically partitioned sources. [5] studied two different formulations of the distributed k-anonymization problem and uses cryptography to obtain provable guarantees of their privacy properties. Our distributed anonymization component falls under this category. But in contrast to the above work, our approach allows data providers to build a *virtual* anonymized database while the data providers still maintain local control of the individual anonymized database. We also address the other end of the problem which is to query the virtual anonymized data.

Secure multi-party computation. The problem of protecting privacy for *data providers* has its roots in the secure multi-party computation (MPC) problem [6]. In MPC, a given number of participants, each having a private data, wants to compute the value of a public function. An MPC protocol is *secure* if no participant can learn more from the description of the public function and the result of function. While there are general secure MPC protocols, they require substantial computation and communication costs and are often impractical for multi-party large database problems. Recently, there have been research focusing on designing specialized secure MPC protocols which are considerably more efficient than applying generic constructions to the same functions [7].

Semantic Integration. An important challenge in data integration is the semantic heterogeneity. The problem of schema matching is to find seman-

tic correspondences (matches) between database schemas. We refer readers to Doan et al. [8] for a recent survey on the most important issues and techniques relevant to schema matching problems.

Privacy-preserving data integration. Clifton et al. [9] were among the first to study the problem of privacy preserving integration and identified a set of key challenges and opportunities. There are a few recent proposals focusing on subproblems, such as secure distributed join [10]. Another related area is privacy preserving data mining across distributed data sources [7]. These works follow the secure multi-party computation model and the main goal is to ensure that data is not disclosed among participating parties while allowing certain mining task to be carried out. Our goal is to build a privacy-preserving data integration system and we expect to leverage many of these techniques in our work, such as incorporating privacy-preserving join in our query processing engine to protect privacy for *data providers* in query execution.

Distributed and federated databases. Distributed database systems have been extensively studied and many systems have been proposed over the years. Earlier distributed database systems [11], such as R* and SDD-1, share modest targets for network scalability (a handful of distributed sites) and assume homogeneous databases. The focus is on encapsulating distribution with ACID guarantees. Later distributed database or middleware systems, such as DISCO [12] target large-scale heterogeneous data sources. Many of them employ a *centralized* mediator-wrapper based architecture

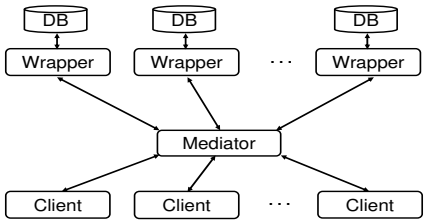

Figure 2: Typical mediator-based architecture.

(see Figure 2) to address the database heterogeneity in the sense that a single mediator server integrates distributed data sources through wrappers. The query optimization focuses on integrating wrapper statistics with traditional cost-based query optimization for single queries spanning multiple data sources. As the query load increases, the centralized mediator may become a bottleneck. Most recently, Internet scale query systems, such as PIER [13], target thousands or millions of massively distributed homogeneous data sources with a peer-to-peer (P2P) or hierarchical network architecture and focus on efficient query routing schemes for network scalability. However they sacrifice on data updates and complex query functionalities and typically relax the consistency guarantee.

While it is not the aim of our system to be superior to any of these works, our system distinguishes itself by addressing an important problem space for querying large-scale heterogeneous and private databases with both network and query load scalability as well as transaction semantics. In spirit, our system is based on a *distributed* mediator-based architecture in which

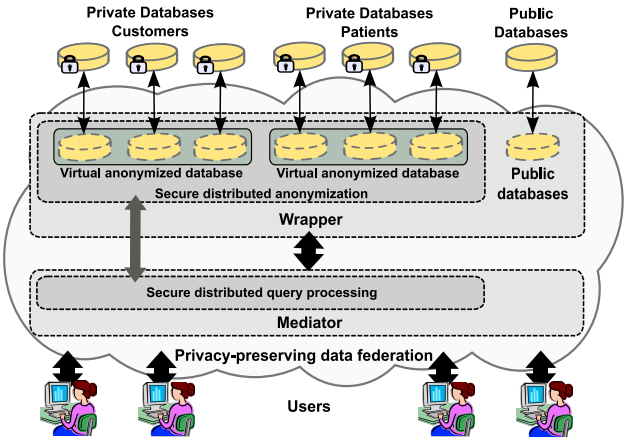

Figure 3: Proposed architecture for scalable and privacy-preserving database federations in the cloud

a federation of mediators and wrappers forms a *virtual system* in a P2P fashion. From scalability point of view, our query processing engine focuses on dynamically placing (sub)queries on the mediators available in the cloud for query-load balancing and scalability.

3 Proposed architecture

In this section, we present our conceptual architecture, describe the underlying technology for the key subsystems, and illustrate a usage scenario through our deployment architecture.

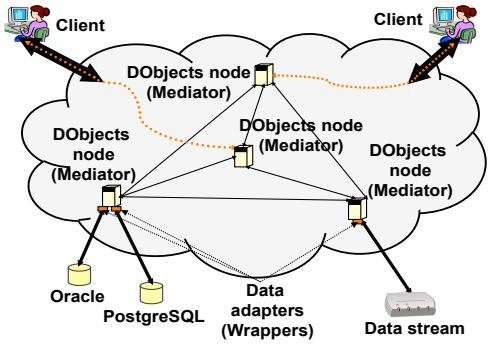

Figure 4: Deployed architecture for distributed querying infrastructure

3.1 Overview

The proposed conceptual architecture of our *micro-cloud* is illustrated in Figure 3. It employs a mediator-based architure for federating distributed heterogeneous data sources. The wrapper layer is responsible for retriving data from individual data sources. The mediator layer is responsible for mediating queries spanning across multiple data sources, routing subqueries to wrappers, and aggregating the results. To address the privacy issues for data sources that contain personal data, 1) the *wrapper* layer employs a *secure distributed anonymization* engine that builds a *virtual* anonymized view of the data while preserving privacy for both *data subjects* and *data providers*, and 2) the mediator layer employs a *secure distributed query processing* engine to aggregate results from multile data sources. Importantly, users do not need to be aware of the fact that data is distributed or private and secure anonymization and query processing engines are employed automatically and

transparently if a query involves private data.

A novel aspect of our architecture is that it is implemented on top of a *metacomputing* framework we have developed called DObjects [14]. It consists of multiple decentralized system nodes which can serve as wrappers and/or mediators and form a *virtual system* in a P2P fashion. Figure 4 depicts DObjects framework deployed for the cloud. The system has no centralized services and uses the *metacomputing* paradigm as a resource sharing substrate to benefit from computational resources available in the cloud. Each node in the system can be considered a *droplet* as it provides similar functionality to other nodes and all the droplets form a *micro-cloud*. Each droplet can serve as a *mediator* that provides its computational power for query mediation and results aggregation. Each droplet can also serve as a data adapter or *wrapper* that pulls data from data sources and transforms it to a uniform format that is expected while building query responses. Users can connect to any system node; however, while the physical connection is established between a client and one of the system nodes, the logical connection is established between a client node and a virtual system consisting of all available nodes, or the *micro-cloud*. We discuss the two major tiers of the system, the wrapper and the mediator, in subsequent subsections.

3.2 Wrapper

The wrapper layer is responsible for retrieving data from individual data sources and forming a virtual database. We assume that public and private

databases coexist in the system. Public databases are simply part of the virtual database. Private databases with personal data, on the other hand, need to have their data anonymized before contributing it to the virtual database. In addition to support single database anonymization, we support a distributed data anonymization scheme.

Distributed anonymization. In our distributed anonymization approach, data providers participate in distributed protocols to produce a *virtual* integrated and anonymized view of their data. In order to protect the privacy for *data subjects*, our current protocol is based on the Mondrian algorithm [3] that uses greedy recursive partitioning of the (multidimensional) quasi-identifer domain space to satisfy k-anonymity for the data subjects.

In order to protect the privacy for *data providers*, our problem can be viewed as designing secure multi-party computation protocols for anonymization (building virtual anonymized view) and later query processing (assembling query results). The key idea for the distributed anonymization protocol is to use a set of secure atomic multi-party protocols to realize the Mondrian method for the distributed setting [15].

Important to note is that in our approach, each database produces a local anonymized dataset that still resides at individual databases. Individual local anonymized dataset is not required to be k-anonymous by itself, however, their integration forms a *virtual* database that is guaranteed to be k-anonymous. When users query the virtual database and get routed to individual wrappers, they execute the query on the local anonymized dataset,

and then engage in a distributed protocol to assemble the results that are guaranteed to be k-anonymous. We will discuss the issue of how to query the virtual anonymized data in the subsection below.

3.3 Mediator

The mediator layer is responsible for mediating and decomposing queries, and assembing query results from individual data sources.

Schema integration. The role of schema integration subsystem is to address the issues of database heterogeneity. In our current work, we do not focus on schema matching problem. We assume that the global mediated schema is defined by system administrators in the system configuration. For systems with a handful DObjects nodes (the number of data sources can be still large), the configuration can be replicated and synchronized at every node as the cost of synchronization will be relatively small. For larger scale systems with more DObjects nodes, the global schema can be replicated at a subset of the DObjects nodes such as landmark nodes.

Distributed query execution. Query execution is the main functionality of the mediator layer, with the goal of parsing, efficient deployment and execution of queries. Due to the fact that the mediator is distributed, new issues and opportunities arise during query execution. (Sub)queries can be migrated from one node to another to speedup execution, increase system throughput or decrease load of particular parts of the system. The migration

is both network-aware and load-aware.

The query execution component provides two major types of database operators: classical and secure. The classical operators include well-understood query operators like selection, projection, join and distributed join. The secure operators are *integrated* into our query processing engine to handle anonymized views of private data sources. We consider this as one of the main contributions of our system.

The secure operators are designed to protect privacy of *data providers* when querying the virtual database. For instance, in our current implementation, the distributed anonymization protocol discussed above enables a group of nodes to produce a virtual k-anonymous database based on the union of the data horizontally split among the nodes. The local anonymized datasets are not necessary k-anonymized. However, the union of the data forms the virtual database and is guaranteed to satisfy k-anonymity requirement. When a query is received, individual wrappers run the query against its local anonymized dataset, and the results are integrated (unioned in this case) using a *secure union* protocol to protect privacy or anonymity for the participants.

It is also worth mentioning that the predicates specified in the query for anonymized data will have different semantics than those specified for the non-anonymized databases. In case of the anonymized data the predicates restrict results to tuples that *possibly satisfy query predicates*. The reason for such behavior is that anonymization can remove or generalize exact values of

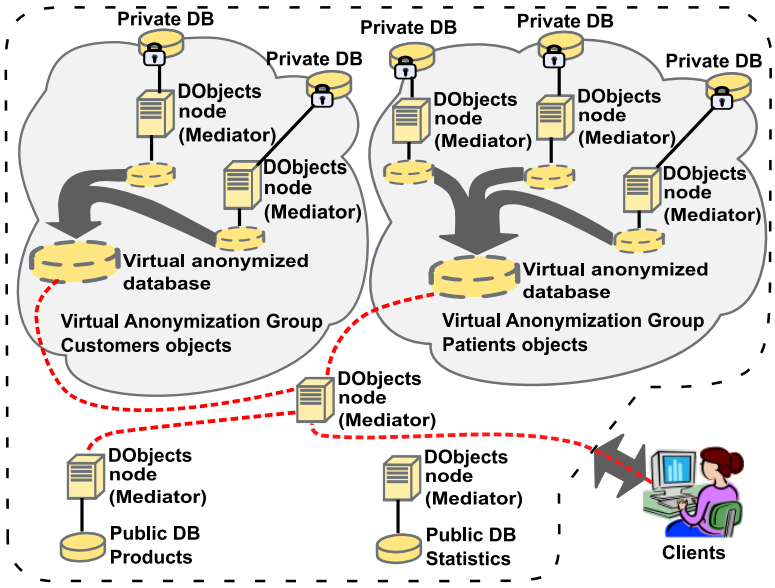

Figure 5: Deployment architecture for scalable and privacy-preserving database federations.

attributes and any predicate evaluation has to be performed after the data is anonymized.

3.4 Deployment Architecture and Scenario

Having described the conceptual architecture and the key subcomponents of our system, we now present a deployment architecture for providing privacy-preserving data federations services, or a *micro-cloud*, using our DObjects framework through an examle scenario. We consider a scenario with four sources of data: a group of cooperating hospitals with patients databases, a group of stores with customers databases, products databases that contain products sold in the stores and statistical database that provides some

statistical information.

The vision of the deployed system is presented in Figure 5. The key point of the *micro-cloud* is that it employs DObjects nodes which utilize resources and data available in the cloud and form a virtual distributed system that can be used by clients. Despite the fact that physically the system is distributed, the virtual layer provided by DObjects offers clients an abstraction of centralized system that can be used by submitting queries in the form of SQL. The goal is to provide a seamless access to the virtual database consisting of publicly available data as well as private or personal data.

Access to data requiring distributed anonymization is provided through a *virtual group*. DObjects nodes are capable of forming a peer-to-peer ad-hoc collaboration groups that will work on designated tasks. For instance, in case of distributed anonymization, such a virtual group consists of all the nodes that provide private data of given type (e.g. all the nodes that provide Patient or Customer data should collaborate in a virtual group). The role of the nodes in such anonymization virtual group is twofold. First, the nodes participate in a distributed anonymization protocol that builds a distributed anonymized view of data (the anonymized database is still distributed though). Second, during query execution, when secure union operator is used to access the distributed anonymied database created before, those nodes participate in a secure union protocol. Note that the virtual group working on distributed anonymization is defined by the data placement.

```
select * from patients where ZIP in
   (select ZIP from Customers c  where c.item in
      (select id from Products p
                 where p.name like "milk%"))
```

Figure 6: Sample query.

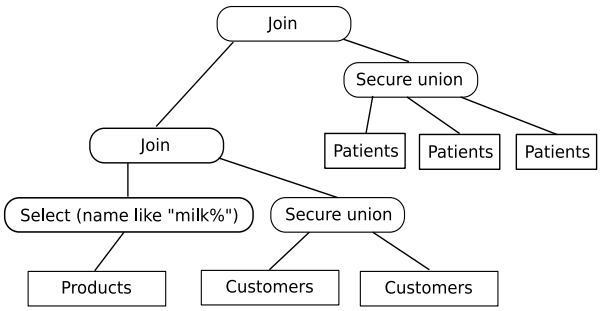

Figure 7: Sample query plan.

When a DObjects node receives a query, it performs an initial query decomposition. Parts of sub-queries that do not involve secure objects (i.e. objects that require anonimization) can be simply answered using classical database operators (joins, selections etc.). On the other hand, sub-queries that involve any of the complex anonymization will be routed to virtual groups. Note that the anonymization algorithm needs to be run only once, probably when a virtual group is formed for the first time. Any further queries submitted to the group will not involve full anonymization, but only query execution part that unions the results in a way that protects security of data owners. An interesting fact is that once the data is obtained from a virtual group that performs anonymization, it can be later processed by any

standard database operator. For instance, when anonymized patients need to be joined with another objects, a standard implementation of distributed join operator can be used.

Now let's consider a sample query as presented in Figure 6. The query selects all the patients that have purchased products with name starting with "milk". A possible query plan that will be generated by the query infrastructure is depicted in Figure 7. Please note that classical database operators (joins, selections, projections) are used together with a new operator, secure union. Secure union is used to access virtual anonymized patients and virtual anonymized customers databases. Note that the secure union operators will be used in a transparent way from users' point of view, and no special consideration is required while building the queries.

3.5 Implementation Status

At the present time we have implemented all the components of the basic DObjects framework[7]. The basic mechanisms within this framework, including dynamic and iterative query optimization and execution protocols and transaction management, have been experimentally evaluated [14]. We have also implemented the distributed anonymization and the secure query operator based on k-anonymity approach and integrated it into DObjects. We performed some preliminary evaluations [15] and the results have demon-

[7]The basic DObjects framework is available for download at http://www.mathcs.emory.edu/Research/Area/datainfo/dobjects/

strated that distributed anonymization protocol providers much better data utility than the naive independent anonymization approach.

4 Summary

We have presented a system that enables privacy-preserving data sharing for cloud computing. Our system enables transparent access to data and resources available in the cloud. Sensitive data is anonymized using distributed anonymization algorithms, and the system provides transparent access to these anonymized data.

Our work continues along several directions. First, we are planning to implement the fully functional prototype of secure data sharing platform based on DObjects data services. Second, we are planning to investigate other anonymity principles and algorithms and evaluate their usefulness for distributed anonymization. Finally, we are planning to look into the problems of schema matching to provide support for administrators in this area of the configuration.

References

[1] Logothetis, D., Yocum, K.: Ad-hoc data processing in the cloud. Proc. VLDB Endow. **1**(2) (2008) 1472–1475

[2] Sweeney, L.: k-anonymity: a model for protecting privacy. International journal on uncertainty, fuzziness and knowledge-based systems **10**(5) (2002)

[3] LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: IEEE ICDE. (2006)

[4] Jiang, W., Clifton, C.: A secure distributed framework for achieving k-anonymity. The VLDB Journal **15**(4) (2006) 316–333

[5] Zhong, S., Yang, Z., Wright, R.N.: Privacy-enhancing k-anonymization of customer data. In: PODS. (2005)

[6] Goldreich, O.: Secure multi-party computation (2001) Working Draft, Version 1.3.

[7] Clifton, C., Kantarcioglu, M., Lin, X., Vaidya, J., Zhu, M.: Tools for privacy preserving distributed data mining (2003)

[8] Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine **26** (2005) 83–94

[9] Clifton, C., Kantarcioğlu, M., Doan, A., Schadow, G., Vaidya, J., Elmagarmid, A., Suciu, D.: Privacy-preserving data integration and sharing. In: DMKD '04: Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, New York, NY, USA, ACM (2004) 19–26

[10] Agrawal, R., Asonov, D., Kantarcioglu, M., Li, Y.: Sovereign joins. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering, Washington, DC, USA, IEEE Computer Society (2006) 26

[11] Kossmann, D.: The state of the art in distributed query processing. ACM Comput. Surv. (2000)

[12] Tomasic, A., Raschid, L., Valduriez, P.: Scaling heterogeneous databases and the design of disco. In: Proc. of the ICDCS. (1996)

[13] Huebsch, R., Chun, B.N., Hellerstein, J.M., Loo, B.T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I., Yumerefendi, A.R.: The architecture of pier: an internet-scale query processor. In: CIDR. (2005)

[14] Jurczyk, P., Xiong, L., Sunderam, V.: DObjects: Enabling distributed data services for metacomputing platforms. In: Proc. of the ICCS. (2008)

[15] Jurczyk, P., Xiong, L.: Privacy-preserving data publishing for horizontally partitioned databases. Technical Report TR-2008-013, Emory University, Math&CS Dept. (2008)