DObjects+: Enabling Privacy-Preserving Data Federation Services

Pawel Jurczyk [*1], Li Xiong [#2], Slawomir Goryczka [#3]

[*]*Google Inc.*
Cambridge, MA, USA
[1]pjurczy@gmail.com

[#]*Department of Mathematics and Computer Science, Emory University*
Atlanta, GA, USA
[2]lxiong@emory.edu
[3]sgorycz@emory.edu

***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, it remains a challenge to ensure the privacy, interoperability, and scalability for such services. In this paper we demonstrate a scalable and extensible framework that is aimed to enable privacy preserving data federations. The framework is built on top of a distributed mediator-wrapper architecture where nodes can form collaborative groups for secure anonymization and secure query processing when private data need to be accessed. New anonymization models and protocols will be demonstrated that counter potential attacks in the distributed setting.**

I. Introduction

With the trend of cloud computing, data and computing are moved away from desktop and are instead provided *as a service*. There is an increasing need to provide data-as-a-service with the goal of facilitating access to a wealth of data across distributed, heterogeneous and private data sources. 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.

While the scenario above demonstrates the increasing needs for integrating and querying data across distributed and autonomous data sources, it remains a challenge to ensure privacy, interoperability, 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 as if it was a part of a single system.

Research challenges. There are two important privacy and security constraints to be considered for data federation services: 1) the privacy of individuals or *data subjects* (such as patients), and 2) the security of *data providers* (e.g. institutions) who want to protect their data or data ownership. The problem of protecting privacy for the data federation environment is especially challenging due to the distributed data providers. Protecting individual privacy for a *single* database (in client-server setting) have been extensively studied in recent years [4] and the security of data providers can be potentially addressed by *secure multi-party computation* (SMC) approaches [13]. However, few works have studied the privacy protection problem for distributed data federation setting. Simply combining the existing privacy models in single provider settings and security notions in multi-party computation setting is not sufficient in addressing the potential complexities in distributed anonymization. Concretely, the data distribution at mutually untrusted data providers introduces new privacy threats caused by data providers in addition to those caused by data recipients, which have not been studied in any of the existing privacy models. In addition, the anonymized results, while protecting privacy for individuals, may reveal certain ownership of the data for data providers. Table 1 list the new challenges or potential attack space introduced in the data federation environment that are not applicable or have not been previously considered in single-provider data publishing or SMC setting. Finally, few works have taken a systems approach for developing privacy-enabled data federation middleware for accessing private data in a seamless manner.

TABLE I
Attack Space for Data Federation

		Target	
		data subject	**data provider**
Attacker	**data recipient**	single-provider	new challenge
	data provider	new challenge	SMC

Contributions. We propose to demonstrate DObjects+, a scalable and extensible framework that is aimed to enable privacy preserving data federation services. The framework extends our DObjects architecture [6], [8] with our ongoing work on distributed anonymization protocols [7], [5], [18] and secure query processing protocols [9] for a seamless access to distributed and possibly private data. We summarize the

contributions of the demonstrated framework below.

First, the framework provides a distributed anonymization service for accessing private data. It constructs a *virtual* anonymized database from multiple data providers while preserving privacy for *data subjects* and confidentiality of *data providers*. While the framework is orthogonal to different privacy principles, we studied several representative state-of-the-art privacy principles within our framework including l-diversity [14], t-closeness [12], and differential privacy [3], [10]. We show the implications of adopting them in the distributed setting with respect to the above attack space and integrated new or modified notions and algorithms in our framework [7], [5], [18].

Second, the framework provides a *secure distributed query processing* service for querying the virtual anonymized data in a scalable and secure way. Secure query operators such as secure set union [9] are *integrated* into the query processing engine with the classical query operators. 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 is recognized automatically and transparently from users' point of view and secure operators are deployed to guarantee privacy of the data.

Finally, the framework is built on top of a *distributed* mediator-wrapper architecture [6] where individual system nodes serve as mediators and/or wrappers. The architecture is flexible and extensible in that the nodes form collaborative groups for secure anonymization and secure query processing when necessary. Therefore, it provides seamless privacy preserving data federation that can be easily deployed in the cloud. 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* form a *micro-cloud*. In spirit, the data federation framework we demonstrate can be considered as such *micro-cloud*.

We realize, sharing the vision and insights from [2], [1], that there are many open issues and alternative approaches in building a full-fledged and integrated privacy-preserving data federation system. As the first attempt, we focus on the middleware framework with a selected set of privacy principles and query processing protocols. We hope to demonstrate the feasibility of the framework and the possibility of using it as a platform for integrating and evaluating further developed models and protocols.

II. Related Work

Earlier distributed database systems [11] share modest targets for network scalability (a handful of distributed sites) and assume homogeneous databases. Later distributed database or middleware systems (e.g, DISCO [17]) target large-scale heterogeneous data sources and use a *centralized* mediator-wrapper based architecture. Our system is based on a *distributed mediator* architecture in which a federation of mediators and wrappers forms a *virtual system* in a P2P fashion and hence provide a scalable and extensible architecture for integrating privacy preserving technologies in a seamless fashion.

The problem of protecting privacy for *data subjects* in the released data for a single database has been extensively studied in recent years [4]. Most literature following the seminal work on k-anonymity [15], [16] and l-diversity [14] adopts an adversarial or Bayes-optimal privacy notion [14]. Differential privacy [3] is emerging as a strong notion that guarantees privacy with arbitrary background knowledge. However, the implications of these privacy principles for mutually untrusted distributed data providers have not been carefully studied.

The problem of protecting privacy for *data providers* has its roots in the problem of secure multi-party computation (SMC) [13] in which a given number of participants, each having a private data, wants to compute the value of a public function. Our problem can be viewed as designing SMC protocols for anonymization and query processing. In addition to leveraging existing SMC protocols for subroutines, our research also requires careful design of the (centralized version of) anonymization algorithms and development of efficient protocols for online query processing. Clifton et al. [2] were among the first to study the problem of privacy preserving integration and identified a set of key challenges and opportunities. Bhowmick et al. [1] proposed an interesting conceptual architecture but the architecture was fairly complex and there was no system built.

III. Framework

A. Architecture and Overview

Our framework employs a distributed mediator-based architecture illustrated in Figure 1. Logically, it consists of two layers: mediator and wrapper. Physically, it consists of multiple decentralized system nodes which can serve as a mediator and wrapper (M/W) or a mediator (M) and form a *virtual system* in a P2P fashion. The wrapper is responsible for retrieving data from individual data sources, while the mediator is responsible for mediating queries spanning across multiple data sources, routing subqueries to wrappers, and aggregating the results.

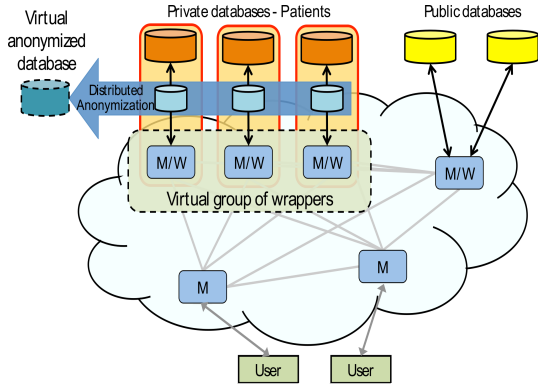

Fig. 1. Architecture for privacy-preserving data federation.

The framework employs two integrated components to address the privacy and confidentiality constraints: 1) *secure distributed anonymization* at the *wrapper* layer that builds a *virtual* anonymized view of the data while preserving both privacy for *data subjects* and confidentiality of *data providers*, and 2) *secure distributed query processing* at the *mediator* layer to securely aggregate results from multiple data sources.

Public and private databases can coexist in the system. The system nodes form collaborative groups for secure anonymization and secure query processing when necessary and therefore provide seamless privacy preserving data federation from users' point of view.

B. Distributed anonymization

In distributed anonymization, data providers participate in distributed protocols to produce a *virtual* integrated and anonymized view of their data. Important to note is that in our approach, each database produces a local anonymized dataset that still resides at individual databases. Given a privacy principle, individual local anonymized dataset may not satisfy the principle itself, however, their integration forms a *virtual* database that is guaranteed to satisfy the given privacy principle. When users query the virtual database and the query gets 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 satisfy the privacy principle.

Below we briefly present the representative privacy principles and protocols we have studied and implemented within our framework. We note that our framework is independent of specific privacy principles and our focus is to demonstrate the implications of adopting them in a distributed setting.

Adversarial privacy. As we have discussed earlier, when we adopt an adversarial privacy principle such as l-diversity or t-closeness with generalization or grouping techniques for distributed data providers, we have to consider: 1) additional privacy threats for data subjects caused by data providers, and 2) confidentiality threats for data providers caused by published data. We briefly discuss below our approach in addressing them.

Privacy threats caused by data providers. Since the data providers are mutually untrusted, each data provider, with the knowledge of its own data records (in addition to the assumed background knowledge of the given principle), may compromise the data records at other publishers. It becomes worse when data providers collude with each other. We propose a general distributed adversarial privacy notion, m-privacy, which specifically models an attacker consisting of up to m colluding providers [5]. An anonymization satisfies m-privacy with respect to a given privacy constraint C such as l-diversity, if and only if the records in each group excluding any subset of records from any m data providers satisfies the constraint C. Essentially, this definition considers any combination of up to m colluding providers as an m-adversary, and guarantees that the records in each group excluding any of those from the attacker still satisfies C. Our framework currently uses a greedy partitioning approach and an efficient algorithm for checking the constraints for all possible combinations of m-adversaries.

Confidentiality threats for data providers. Given certain background knowledge, the anonymized data, while protecting the sensitive attributes of a data subject, may reveal the ownership of certain records by a data provider. We defined a new notion, *l-site-diversity*, as one solution to protect the data ownership anonymity for data providers as well as a generalization based protocol [7]. Similar to l-diversity in spirit, the *l-site-diversity* requires that records in each anonymous group has to belong to at least l "distinct" providers. This notion protects the anonymity of data providers (instead of data subjects) in that each record can be linked to at least l providers.

Differential privacy. Differential privacy [3] supports *statistical* data release and makes no assumption on the background knowledge of malicious data users. It requires the result of a given computation to be formally indistinguishable when computed with and without any particular record in the dataset, as if it makes little difference whether an individual is being opted in or out of the database. This notion is extremely advantageous in the distributed setting where multiple data providers can use their own data as background knowledge and collude with each other, which has to be explicitly modeled when using adversarial privacy principles. To support non-interactive data release with differential privacy, our framework currently uses an adaptive multi-dimensional partitioning strategy for releasing histograms and an optional set of synthetic data generated from the histograms [18].

C. Distributed query processing

The query processing component located in the mediator provides two major types of database operators: classical and secure. The classical operators include well understood query operators like selection, projection and join. The secure operators are *integrated* into our query processing engine to handle anonymized views of private data sources.

The secure operators are designed to protect security of *data providers* when querying the virtual database. For instance, the distributed anonymization protocol discussed above enables a group of providers to produce a virtual anonymized database based on the union of the data horizontally split among them. When a query is received, individual wrappers run the query against its local anonymized dataset, and the results are integrated using a *secure union* protocol to protect confidentiality or anonymity for the participants.

An example query is presented in Figure 2 and the corresponding query plan is presented in Figure 3. Note that secure and classical query operators are used as needed, without user intervention.

```
select c.name, p.age, f.number
from DemographicInformation c,
     c.lPatients p, c.lFlights f
where c.name like "%san"
```

Fig. 2. Query example (Object Query Language).

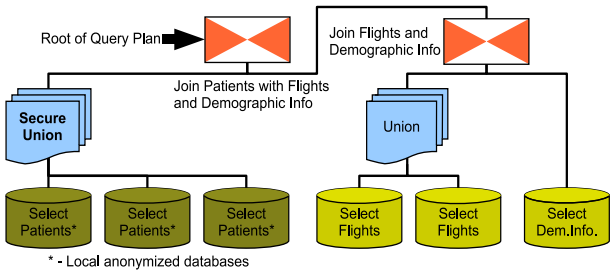

Fig. 3. Sample query plan.

IV. Demonstration

In the demonstration, we will show the functionalities of our implemented system[1], highlighting key aspects of distributed anonymization.

Setup. The demonstration setup will use a client machine in the demonstration room and will connect to a set of physical machines (5-10) located at Emory University. All the remote nodes will act as DObjects nodes (mediators). A subset of the nodes will have data stored in their local databases and will act as data wrappers.

Through a user interface installed on the client machine, we will issue queries that involve both private personal data and public data. The user interface will allow users or audience to issue queries, examine query results, as well as examine logging information and analyze how the decisions in the distribution anonymization protocol are made and how the secure query operators are executed.

The data setup will follow the example scenario of the healthcare domain and will include the following object types: Air and Rail Connections, Demographic Information and Patients. The first two objects will be horizontally partitioned across a few databases. Patient objects will also be partitioned and will require anonymization to protect their sensitive data.

Highlights. The demonstration will focus on the following aspects: 1) basic data operations of the system including query building and results retrieval (both anonymized and non-anonymized data), 2) configuration of groups of nodes building virtual anonymized databases, 3) comparison of the results using different anonymization protocols with different privacy principles, and 4) performance and scalability of the system in terms of distributed anonymization and secure query processing.

Audience will be able to issue queries through the user interface involving different data sources and examine the local databases before and after anonymization as well as the final results. An example query is shown in Figure 2. During the demonstration, we will change the configuration of groups used for distributed anonymization a few times. The demonstration interface also offers a look at logging information so the audience will be able to see under-the-hood of the distributed anonymization steps and query execution plan as shown in Figure 3.

As performance is often the concern for a distributed system, we will show the performance overhead of running various queries. The performance overhead in our framework is impacted by two factors: the anonymization and secure query processing. Our analysis and evaluations on the overhead of particular distributed anonymization algorithms [7] and secure protocols such as set union [9] in our framework show that they are efficient and scalable. In the demonstration, we will show the performance overhead of the overall query processing for several queries involving different combination of private and public data.

Backup plan. In case the demonstration room does not provide connectivity with our system nodes deployed at Emory University, our backup plan is to start a few system nodes on the local machine and our demonstration application will connect to these local nodes. We will also run simulations of various settings to demonstrate the results of our distributed anonymization.

[1] http://www.mathcs.emory.edu/Research/Area/datainfo/dobjects

Acknowledgement

This research was supported in part by a Cisco Research Award and NSF grant CNS-1117763. The authors would like to thank the anonymous reviewers for their suggestions that helped improve this demo.

References

[1] S. S. Bhowmick, L. Gruenwald, M. Iwaihara, and S. Chatvichienchai. Private-iye: A framework for privacy preserving data integration. In *ICDE Workshops*, page 91, 2006.

[2] C. Clifton, M. Kantarcioğlu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, and D. Suciu. Privacy-preserving data integration and sharing. In *DMKD workshop at SIGMOD*, pages 19–26, 2004.

[3] C. Dwork. Differential privacy: A survey of results. In *TAMC*, volume 4978 of *Lecture Notes in Computer Science*, 2008.

[4] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. *ACM Com Surveys*, 42(4), 2010.

[5] S. Goryczka, L. Xiong, and B. C. M. Fung. m-Privacy for collaborative data publishing. In *CollaborateCom*, 2011.

[6] P. Jurczyk and L. Xiong. DObjects: enabling distributed data services for metacomputing platforms. *VLDB*, 1(2), 2008.

[7] P. Jurczyk and L. Xiong. Distributed anonymization: Achieving privacy for both data subjects and data providers. In *DBSec*, 2009.

[8] P. Jurczyk and L. Xiong. Dynamic query processing for p2p data services in the cloud. In *DEXA*, 2009.

[9] P. Jurczyk and L. Xiong. Information sharing across private databases: Secure union revisited. In *IEEE PASSAT*, 2011.

[10] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 international conference on Management of data*, SIGMOD, 2011.

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

[12] N. Li and T. Li. t-Closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, 2007.

[13] Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1), 2009.

[14] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-Diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3, 2007.

[15] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6), 2001.

[16] L. Sweeney. k-Anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[17] A. Tomasic, L. Raschid, and P. Valduriez. Scaling heterogeneous databases and the design of disco. In *ICDCS*, 1996.

[18] Y. Xiao, L. Xiong, and C. Yuan. Differentially private data release through multidimensional partitioning. In *Proceedings of the 7th VLDB conference on Secure data management*, SDM, pages 150–168, 2010.