Indian Institute of Management Calcutta

Working Paper Series

WPS No. 763
June 2015

Information Retrieval as a Service – IRaaS : A Concept Paper on Privacy Analysis

Asim K. Pal
Professor, Management Information Systems Group
Indian Institute of Management Calcutta
D. H. Road, Joka, P.O. Kolkata 700104, India
http://facultylive.iimcal.ac.in/workingpapers
asim@iimcal.ac.in

S. Bose
Professor, Head Dept. Computer Science and Engineering
Neotia Institute of Technology, Management and Science
D. H. Road, Jhinga, PO: Amira, South 24 Parganas, WB–743368, India
subratabose@yahoo.com

Abstract

Privacy analysis has not always got proper attention in the literature often overtaken by security algorithms. This work attempts to fill in this gap. The strength of this work we believe is in the privacy analysis conducted in depth for a complex problem following an objective method. Providing information retrieval service from multiple heterogeneous autonomous data sources is a natural requiremel(Q0 .7(l)5f2.5(qu)-l8wi)u0o035.3(multipe.l)6.57.6(r)ions [e]earqulel(Q0 3(a)9.4(8win

this as

privacy concerns of the participants (see section 1.6 below). But there does not yet seem to be any concerted attempt to *holistically* look into the problem state of technology we believe that CA-IRaaS is more feasible and practical at this point of time. Our privacy model concerns this information service.

The paper is organized as follows. Section 1.1 to 1.3 discusses the importance, feasibility and rational behind IRaaS being a cloud based service, the role of mediator for IRaaS is discussed at section 1.4, the privacy need of IRaaS is discussed at section collaboration capability of the cloud could be another reason. But as such there is no pressing need for IRaaS to be cloud based if it is meant for a simple application. In our way of th same time. Proprietary nature of existing cloud service providers restricts consumers to use multiple cloud services simultaneously for the same problem. Collaborative cloud computing for software services enables customers to have better access to software, computing facilities, and data and also create more business opportunities [31, 42]. For example, a snapshot of customer's data from various data sources would help a user access to information which would have otherwise been difficult for him to assimilate. This could be as open as train or flight information or as restricted as financial records or crime records, etc. With democratization and collaborative cloud computing information can be obtained dynamically as per the arrangement and need of the business. Yoon et al. [42] presents a mathematical model for dynamic collaboration of cloud service providers for auction market to offer collaborative services to its customers. Formation of the collaborators is initiated by one of the providers who act as a primary CP to form a virtual organization with other collaborators for providing a set of services to its customers. Karnouskos et al. [43] proposed a SOA based service architecture for industrial automation. The proposed architecture will offer a collection of services providing common functionalities, interact with each other and form a cloud of services which need to be collaborative. Query executions in a collaborative cloud [39] in which different parties need to release information and cooperate with others require protection of sensitive information. The data source participating in such systems could be completely independent, federated or a centrally planned distributed database system. Query processing in such a scenario should support selective sharing of information by different data owners (similar to restrictive view, authorizations and access restriction mechanisms in relational databases) as per their access authorization to different players. The problem thus requires a solution that helps capture different data protection needs of the cooperating parties. S. Vimercati et al. [44] presents an approach for the specification and enforcement of authorizations regulating data release among data owners collaborating in a distributed computation, to ensure that query processing discloses only data whose release has been explicitly authorized. The authors also present an algorithm that determines whether a given query plan can be safely executed and if so produces a safe execution strategy. Answering queries with access restrictions has been studied extensively in the literature [45].

1.4 Mediator for IR service

Let us now focus on the job of mediation performed by an IRaaS provider. We have to assume that the service provider is adequately knowledgeable about the data sources required and resourceful and trustworthy to connect them. It is very much possible that the service provider looks for appropriate data sources by using his or her contacts, by searching through the net, or inviting for participation (possibly through a bidding process), etc. Data sources would join the provider depending on their interests, their knowledge about the provider and also based on the amount of trust they have on the provider and finally establish a business deal with the provider on revenue sharing and pricing schemes, etc. Ultimately, a list of data sources (information providers) becomes part of a given IR service. But this list will occasionally change, depending on entry of new sources or exit of old sources. Having established the data sources the service provider collects meta information about the exposable data of each data source. The data could be heterogeneous in a number of ways, the content of data (text, audio, video), formatting of individual data elements, and data structure (e.g. flat protect their identity and respective assets from each other including SP and thus call for privacy concern among them. Privacy and security of cloud systems for anonymous communication have a centralized or semi-centralized architecture, including Anonymizer, AN.ON, Tor, Freedom, Onion Routing, and I2P.

1.6 Related work

Providing IR service from the data owned by different independent and autonomous data sources demands integration of heterogeneous data lying in multiple servers. A number of approaches have been proposed in the literature, mediator based approach being the most prominent among them heterogeneous multi database system. MD-SQL [26] is a similar work allowing querying data and metadata in a multi database system. The Distributed Interoperable Object Model (DIOM) [4, 6] offers a query mediation framework through an adaptive approach to interoperability instead of an integrated global schema. The DIOM project [6] offers a framework for integration of relational data sources with a centrally performed compilation process [9]. Its main features include information access through a network of *application-specific mediators* which is also aimed for IRaaS implementation. Semantics is an important component for data integration which has led to the inception of ontology-based approach. The pioneering work of Doerr et al. [52] focused on semantic integration and use of ontology for mixing heterogeneous schema across multiple sites. Their efforts have provided a new dimension for information integration.

Privacy is a serious concern in IRaaS and thus privacy preserving

1.7 Contribution

This is a novel attempt to combine the powers of a) cloud computing concept, particularly its SaaS, for scalability and capability to handle complexity, b) distributed computing as a concept for the distributed processing of a complicated information retrieval task, c) data mediation task, d) privacy modelling coupled with security and trust issues to achieve a ubiquitous *Information Retrieval as a Service* for multiple independent and autonomous heterogeneous data sources. We have tried to establish the logic of IRaaS as a cloud based service. A taxonomy has been proposed to suggest two broad categories, Closed Access IRaaS (CA-IRaaS) and Open Access IRaaS (OA-IRaaS). The former one is targeted to a set of applications or an application area where the client data sources are pre-fixed, while the latter one is much more open in its depth and coverage. The taxonomy also delves into collaborative IR services, besides looking into the hierarchy of application areas as the focus of the IR services. Then the work discusses how privacy, security and trust play together a vital role for IR services, mainly for CA-IRaaS. For IRaaS the privacy issues have been discussed at great length, e.g. how different privacy issues are interlinked. A privacy algebra has been suggested to process different privacy issues and privacy protections to enable one to come up with a comprehensive privacy view which is negotiated and agreed across all parties (data sources, the querrier – customer and the service provider). This algebra has been demonstrated on IRaaS. A secure IR framework along with a sketch of the security protocol for IRaaS (including query processing) has been provided. The strength of this work we believe is in the privacy analysis conducted in depth for as complex a problem as IRaaS following an objective method suggested in the work itself. Privacy analysis has not always got proper attention in the literature often overshadowed by security algorithms. This work attempts to fill in this gap.

This paper is based upon Pal et al. [1]. But the current work has added e.g. show me the architectural types of Calcutta during the British rule of India, or show me the most memorable tragic scenes from Charley Chaplin films. These can basically be referred as *Open Access IR Services* (OA íIRaaS), where the mediator has to retrieve data from dynamically

there, e.g. cost sharing or privacy issues. There is still another possibility of application of the idea of CA íIRaaS, which is meant for enterprise applications, enterprise íCA íIRaaS, one instance is for one enterprise, e.g. WM íenterprise íCA íIRaaS for Wallmart, or more narrowly, Mexico íWM-enterprise íCA íIRaaS. It is possible that WM íenterprise íCA íIRaaS is same as USA+EU+Mexico íWM íenterprise íCA íIRaaS, assuming that WM is spread across theses zones. Figure 3 illustrates the taxonomy of an enterprise IR service. A corporate can benefit a lot from such an IR service meant for its own organization, processes, employees, customers, etc by integrating information across the enterprise from heterogeneous data sources. Actually one can think of redesigning their existing ERP systems in view of these kinds of new enterprise based services. And, from the business point of view Enterprise IR services appear to be highly effective for corporate, particularly the big ones. And these can based on the company's private cloud. Further, collaborative cloud computing could be put to good use for developing collaborative IS, both OA and CA types.

Fig 3. Enterprise Closed Access IRaaS for Wallmart

3 IRaaS – the Privacy Issues

In CA-IRaaS model a set of autonomous data owners (data sources) having independent operations allow the *mediator* (the mediating agent on behalf of the service provider who offers the IR service to its information consumers or customers) to access their data based on the query made. In OA-IRaaS model which is more general or flexible a customer makes an arbitrary query for which the IR service provider searches for potential data sources that are relevant to the query and solves the query with the help of volunteering data sources. For the purpose of privacy and security concerns we treat the mediator as an *untrusted third party* (*utp*). We also assume here a *semi honest* or *honest but curious* model for the privacy preserving computation, in the sense that all participating parties would follow the protocol without any deviation but they are free to use any intermediate result or data that pass through them during the execution of the security protocol [7, 8, 13]. The

sometimes the issue of efficiency and cost may overshadow the issue of security. Depending on the complexity of a query it may be beneficial to

Either way maintenance of appropriate data privacy would encourage more data sources participate in the IR service.

d) *Query Privacy* refers to the protection of the customer query from the SP and DSs. The customer is particularly interested to protect the sensitive

Privacy Type	Privacy Protection (Identity)		
	C from DS	DS from C	DS from other DS
0 (Public)	No	No	No
1	No	No	Yes
2	No	Yes	No
3	No	Yes	Yes
4	Yes	No	No
5	Yes	No	Yes
6	Yes	Yes	No
7 (Private)	Yes	Yes	Yes

Table 1: Identity Privacy of Customer and Data Sources (Symmetric Case)

The protection columns indicate whose identity is protected (hidden) from whom. Thus a protection has only two possible values - Yes or No. For the non-symmetric case where each data source decides independently the identity privacy issue is more elaborate. This is expressed as follows:

Privacy Protection (Identity)								
C from DS_1	. . .	C from DS_N	DS_1 from C	. . .	DS_N from C	DS_1 from DS_2	. . .	DS_N from DS_{N-1}
*	*	*	*	*	*	*		*

Table 2: Identity Privacy of Customer and Data Sources (Non-Symmetric Case)
[wildcard * indicates either "Yes" or "No"]

The table shows the existence of :0 E;6 privacy protections, which implies t$^{Ç:Ç>5}$ privacy types. In other scenarios, for example, there may be some public data sources which do not mind incoming communications for any given query. Further note that the privacy statement may change from query to query.

Let us next consider Schema privacy. Here the main concerns are protecting individual schemas S_k of the data sources &5 G L s å0being protected from other DSs as well from the customer. This is because the SP already has the knowledge of schemas. Thus we have the following possible privacies:

Privacy Type	Privacy Protection (Schema)

data sources does not reveal much information to any individual DS regarding other DSs, unless the number of DSs is not too small.

Privacy Type	Privacy Protection (Result)	
	Final Result - SP Protected From	
	C (for intermediate result)	Any of the DSs (for final result)
0 – 3	*	*

Table 7: Result Privacy for SP (Symmetric for Data Sources)

Query privacy refers to the protection of the full query Q (originated from C and passed onto SP with possible encryption of sensitive parts) and $3_{Þ}$ á G L s á ä ä á J (n = number of DSs finally selected by SP for query solving) belonging to $\& 5_{Þ}$ $3_{Þ}$ may or may not have a sensitive part. C would like to protect Q from SP in the sense that it would not like the sensitive parts of Q are disclosed to SP through implication of any information passing through it. C also might like to protect $3_{Þ}$'s (sensitive parts) being protected from $\& 5_{Þ}$ Here the symmetry between the data sources is very much expected both from SP and C's points of view.

Privacy Type	Privacy Protection (Query)
	Qk Protected From DS $_{k}$
0	No
1	Yes

Table 8: Query Privacy of Data Source DSk

Privacy Protection (Query)		
Privacy Type	C Protected From	
	SP	Any of the DSs
0 – 3	*	*

Table 9: Query Privacy for SP (Symmetric for Data Sources)

Finally we come to what is known as Query distribution privacy. This is a totally different kind of privacy. The issue is while a query is being executed through a collaborative computation undertaken by a set of data sources, the SP and also possibly the customer a simple knowledge can greatly influence how the security is implemented as well as how the efficiency is going to be achieved. Usually, there will be a tension between these two factors, though security takes a priority in most situations. This knowledge is regarding the choice of the set of $\& 5_{ȝ}$ á ä $ä_{á}$ á & 5 made by the SP and mutually agreed by all (the customer usually wouldn't be involved in this process.). We are talking about the disclosure of identities of the chosen DSs – we call this *Query Distribution* knowledge. This disclosure can be made to the DSs and / or to C. The most restrictive one would be when it is not disclosed to either of them, we call that *Closed Query Distribution* (*Closed Qd*). This seems to be the most acceptable privacy as it makes preservation of privacy much simpler. The other options are named *Data Source Open Query Distribution* (*DS-Open Qd*), *Customer Open Query Distribution* (C-Open

Qd) and *Open Query Distribution* (*Open Qd*) depending on whether the knowledge is made open to the DSs (all of them – it doesn't make sense to distinguish one from another), C or both. Openness helps in query efficiency but makes it harder to ensure privacy. If data sources are public, the open schemes would be more useful. The privacies are put down in the following table.

Privacy Protection (Query Distribution)		Privacy Type
SP Protected From		
C	Any of the DSs	
Yes	Yes	Closed Qd
Yes	No	C-Open Qd
No	Yes	DS-Open Qd
No	No	Open Qd

Table 10: Query Distribution (Qd) Privac y for SP (Symmetric for Data Sources)

4 Interdependence of the Privacy Issues

parties to interact. For this we have developed algebra based on *join* and *dominance* relation between privacy issues and protections. Finally we look into some feasible scenarios.

The basic idea behind this algebra is the simple fact that a privacy issue need not be completely independent of other issues. For example, if identity of a data source is protected from the customer then the customer cannot access the data source for its schema or data. Thus the identity protection automatically gives protections to other type of privacies. We envisage that identity privacy dominates schema privacy and data privacy for protection of DS from C. Again if we examine separately we find schema privacy dominates data privacy, query privacy is dominated by identity privacy and so on. This calls for a deep look at the privacy issues against protections and their dominance relations and join. The privacy algebra is built on this idea.

4.1 Privacy Algebra

We develop a simple algebra for constructing *composite* privacy issues and protections from elementary privacy issues and protections. This helps in developing a consolidated model for privacy for a complex multi-party computation.

Definitions:

A *privacy issue* (*entity*) is a specific privacy concern expressed and agreed by all the parties in a multi-party computation. It is represented as a matrix, each column represents a *privacy protection* and row represents a *privacy type*. Let *P* be a privacy type used in the following discussion.

A *privacy protection* refers to the protection of one party, say *a*, from another party, say *b*, i.e. *a* protected from *b*, or conversely, *a* open to *b* w.r.t. the underlying privacy issue and hence it has only two possible values "Yes" (*y*) or "No" (*n*). The set of privacy protections in *P* is denoted by *protection*(*P*).

A *privacy type* refers to a particular combination of protections available in a privacy issue. Sometimes privacy types are labelled for easy reference (e.g. Qd privacy – Table 10). The set of types in P is denoted by *type*(*P*).

P1 is a *type-subset* of *P* if *type*(*P1*) is a subset of *type*(*P*) [use subset notation]. Similarly, *P2* is a *protection-subset* of *P* if *protection*(*P2*) is a subset of *protection*(*P*).

Conditioned Privacy Issue P(*c*) is obtained by applying certain selection condition *c* onto the parent privacy issue *P*, or *P*(*Q*) by imposing another privacy issue *Q* upon it. Note, *P*(*c*) or *P*(*Q*) could be a type-subset, protection-subset or both of P.

A privacy issue having *m* privacy protections has a maximum of $t^{à}$ privacy types. A *non-trivial* privacy issue will have less than $t^{à}$ privacy types. A trivial issue would have 0 or all $t^{à}$ protections.

Examples: Refer to Table 1 displaying the Identity privacy for the Customer and Data Sources for IRaaS (the symmetric case). It has three protections, i) C protected from DS, ii) DS protected from C, and iii) DS protected from other DSs. This privacy has eight types. All or some of the types could be labelled for convenience, e.g. the first type has been called Open or Public – where each party is accessible to other, the last one Closed or Private – where none is accessible to another. Since we haven't put any condition on the issue, there are all 2^3=8 types. From Table 2 one can see that all possible communications are being allowed between any two parties – C and DSs. Thus there are 0:0 E;5 protections and $t^{\text{Ç:Ç>5}}$ privacy types. But note that Identity privacy issue for C and DS (Symmetric Case) is both a type-subset and protection-subset of Identity privacy issue for C and DS (Non-symmetric Case). The condition 'symmetry among the DSs' applied on the latter will reduce it to the former, in other words, the former is a conditioned issue w.r.t. the latter.

$(A.B).C = A.(B.C)$. The join operation is thus *idempotency preserving*

However, the query part would require the knowledge of the respective schema.

Coming to the privacy types, for I > S we have 3 valid privacy types as seen in Table 12a, a crisp form in Table 12b. Similarly for S > D we have 3 valid privacy types as seen in Table 12c, by joining I > S with S > D we have 4 valid privacy types for I > S > D as seen in Table 12d.

I	S	Type
No	No	00
No	Yes	01
Yes	Yes	11

Table 12a: I > S (#type = 3)

I	S	Type
No	No	00
*	Yes	*1

Table 12b:Crisp form of Table 12a

S	D	Type
No	No	00
*	Yes	11

Table 12c: S > D (#type = 3)

I	S	D	Type
No	No	No	000
No	No	Yes	001

problem can be strengthened further by applying this analysis process to different problems such as on line auctions, combinatorial or reverse auctions and on line shopping. This in turn will improve these services as well. As next generation systems will be highly collaborative and will have to share information, interoperability via open communication and standardized data exchange is needed [43]. Such system will need planned privacy model. One such example is collaborative cloud based industrial automation [43].

5 Secure IR Framework

The main task of the information retrieval mediation is to coordinate the communication and distribution of information consumer's query among the mediator, the information consumer and the data sources [6]. Mediator is a software component at middleware layer with the services for information retrieval. The proposed framework of CA-IRaaS is '*central mediator/wrapper*' architecture [9] along with the security mechanism built at the information consumer's end and at the data sources' side. The mediator which sits in between the customer and data sources is basically positioned in SP who provides the necessary interface to the customer for querying. The central mediator contains a universal mediator schema that presents a view of the integrated data to the customers through the application. The mediator architecture is depicted Figure 4.

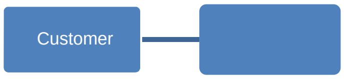

Fig 4: Central Mediator Architecture

The application interface and the central mediator engine are hosted in the Cloud. The mediator engine is interfaced to a number of data sources through wrappers. The central mediator contains a global schema made out of the individual schema of the data sources. Through its application interface IRaaS presents a transparent view of the integrated data to the customer [9]. For each data source there is a wrapper. The wrappers contain code to map the global schema to local schema applicable to individual data source. Customer's query passes through query optimization before mapping by the central mediator and generates query components for each data source. Now, the privacy statement *PS* arrived at through joint negotiation of all the parties involved has to be embedded properly in the algorithm (without privacy considerations). Each action gets modified accordingly.

The system architecture of IRaaS *without privacy mechanism* is summarized in the following steps:

1. Customer sends a query using the IRaaS application interface to the Service Provider
2. The Service Provider accepts the query, determines the set of appropriate data sources to answer the query and hands over the query to the mediator engine
3. Using the global schema the mediator optimizes the query and generates sub query (query components) for individual data sources
4. For data source its wrapper translates the sub query into a query expression that it is executable locally n
5. Each data source executes the sub query and sends the result to the mediator engine through the wrapper
6. At the mediator engine the final result is obtained after joining, selecting or merging as appropriate (if required iterating the process by going back to Step 5) and passes on to the Service Provider
7. The Service Provider returns the answer to the Customer

The system architecture of IRaaS

answer the query (query distribution) is passed on either to the Customer and/or the Data Sources as per the privacy setting.

3. Using the global schema the mediator optimizes the query and generates sub query (query components) for individual data sources.
4. For each data source its wrapper translates the sub query into a query expression that it is executable locally. Depending on the privacy requirement of the query the data source either gets the hidden components directly from the customer or through the Service Provider.
5. After obtaining the sensitive query components each data source executes the sub query and sends the result to the mediator engine through the wrapper. Depending on the specific privacy choice of the customer the query execution may have to be executed differently like PIR where the sensitive components are not even seen at the data source level [15, 27],

Civil-Comp Press, Stirlingshire, UK, Paper 31, 2013. doi:10.4203/ccp.101.31

[2]

[16] S. Hildenbrand, D. Kossmann, T. Sanamrad, C. Binnig, F. Faerber, J. Woehler, "Query Processing on Encrypted Data in the Cloud” by ETH, Department of Computer Science, 2011.

[17] R. Kolavenu, R. Arasanal, "A Survey on Enterprise Databases in Cloud Computing", 2012,

[33]

Autonomous Spontaneous Security. Springer Berlin Heidelberg, 2013. 160-173.

[47] X. Li, S. Goryczka, V. Sunderam, "Adaptive, secure, and scalable distributed data outsourcing: a vision paper." *Proceedings of the 2011 workshop on Dynamic distributed data-intensive applications, programming abstractions, and systems*. ACM, 2011.

[48] W. Shiyuan, D. Agrawal, A. E. Abbadi, "Is homomorphic encryption the holy grail for database queries on encrypted data", *Technical report, Department of Computer Science*, UCSB, 2012.

[49] C. Gentry, "A fully homomorphic encryption scheme.", *Diss. Stanford University*, 2009.

[50] G. Yubin, Z. Liankuan, L. Fengrena, L. Ximing. "A Solution for Privacy-Preserving Data Manipulation and Query on NoSQL Database." *Journal of Computers* 8.6 (2013): 1427-1432.

[51] T. Aditya, S. Chakravarthy, Y. Huang. "Information Integration Across Heterogeneous Sources: Where Do We Stand and How to Proceed?." *COMAD*. 2008.

[52] M. Doerr, J. Hunter, C. Lagoze, "Towards a core ontology for information integration." *Journal of Digital information* 4.1 (2006).

[53] G.Goetz, U. M. Fayyad, S. Chaudhuri. "On the Efficient Gathering of Sufficient Statistics for Classification from Large SQL Databases."*KDD*. 1998.