Collaborative Information Service
Privacy Algebra for User Defined Security

Asim Kumar Pal[1], Subrata Bose[2] and Arpita Maitra[1]
[1]*Management Information Systems Group, Indian Institute of Management Calcutta, Kolkata, India*
[2]*Department of Computer Science and Engineering, Neotia Institute of Technology, Management & Science, Kolkata, India*

Keywords: Privacy Template, Privacy Issue, Privacy Protection, Dominance Relation.

Abstract: With the increased activity over the internet and globalization of the market economy collaborative computing becomes an important area of research. Security is an increasing concern because of chances of malicious elements breaching the network of collaborating partners. Further the level of mutual belief among the collaborators would not be identical and may change with experience. Thus the concept of user defined flexible security requirements arose. The idea of user defined privacy template was brought in IRaaS (Information Retrieval as a Service) (Pal and Bose 2013) which is a flexible system of information services to customers who seek information from various sources through a service provider. The idea was further extended to CIS (Collaborative Information Service) (Pal and Bose 2016) which provides a framework for general information exchange activities (not restricted to retrievals only) for a set of collaborating partners. The current work extends CIS by introducing privacy algebra to be applied on templates to get a concise expression of privacy restrictions. CIS is a step towards a privacy aware collaborative computing problem.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Collaboration is often required by independent organizations for their interlinked business, e.g. partners in a supply chain. Many organizations find that collaboration brings additional value to them. Information exchange or information sharing plays a major role in such joint activities. Consider a typical example from e-commerce. An on-line purchase by a buyer goes through a series of *information activities* such as finding the products of interests, checking their stocks in the warehouses, enquiring delivery status of the items from warehouse to the customer's address, receiving payment through some credit card company, waiting for return of some goods, etc. In another situation, for a bulk order involving different products and shipping addresses all or most of the above tasks could be performed without much of user interaction though the purchase could be on-line or off-line. Such applications can be designed as a *sequence of information exchange activities* which are collaborative in nature. We need efficiency as well as security of the operations among separate and independent organizations or organizational components. This kind of computations comes under what is known as *collaborative computing*. Fundamentally such computation works on the principles of distributed system (Zhu et al. 2006). Security is a serious issue in collaborative computing (Atallah 2006; Ahmed and Tripathi 2007). Security concerns are more serious when the participants do not have much knowledge of each other, e.g. a customer or a small business to a credit card company. Collaboration is successful only when the participants can keep trust in the system. An important security concern in such collaborative computation is at data level. All the data elements belonging to a given participant need not be equally sensitive with respect to specific opponents or its security may need to be traded with efficiency of the activity. This kind of security is known as *attribute oriented security* (Zhang et al. 2006). Security in collaboration can be user oriented or user defined and role based. Pal and Dey (2014) proposed user oriented policies for secure data storage and computation of enterprise data in cloud collecting varying perceptions of users about trustworthiness of the Cloud Service Provider and also roles which the users represent. The work of (Pearson, Shen and

Pal, A., Bose, S. and Maitra, A.
Collaborative Information Service - Privacy Algebra for User Defined Security.
DOI: 10.5220/0005746304730481
In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pages 473-481
ISBN: 978-989-758-167-0
Copyright © 2016 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

Mowbray 2009) has close similarity in terms of offering the user selection of privacy preferences from a range of choices.

In an earlier work Pal and Bose (2013) introduced a robust information retrieval service (IRaaS) to its customer using a service provider from a set of heterogeneous independent and unknown data sources. IRaaS considered four types of *user selectable privacies* – Identity, Data, Query and Result. In a recent work Pal and Bose (2016) introduced Collaborative Information Service (CIS) for secure information exchange among the partners. Like IRaaS, CIS uses *privacy template* to allow users to express their security concerns against inappropriate use of information by the opponents or leakage of information to undesired parties. The current work is an extension of CIS. It proposes *privacy algebra* to represent user defined security captured via privacy template, in a concise and manageable form. CIS caters to seven privacy aspects, namely, Identity, Schema, Data Read, Data Write, Result, Query and Query Distribution. Each of these privacies is looked at the granularity level of each participant against all other resulting in a huge number of combinations. It is not desirable or practically feasible to handle such huge variations which also mean providing a very large number of security mechanisms or protocols. Idea of simplification arose from this. *Privacy algebra* basically helps simplify expressing the overall privacy needs of a collaborative computing job using dominance and join properties of privacies (discussed in section 3).

Rest of the paper is organized as follows. Section 2 discusses CIS. Section 3 describes its privacy model. Section 4 describes privacy algebra. Section 5 shows the applicability of privacy algebra on CIS. Section 6 summarizes the paper and concludes with future direction of the work.

2 CIS SYSTEM

CIS is an information service intended to bring together a number of independent and autonomous entities such as different organizations, departments or units of an organization, individuals etc. for collaborative information exchange among them to perform a task or execute an application. These *entities* form a network, each entity representing a node of the network. The information service processes users' information request which involves data read or write. For every action (query or transaction) a subset of the entities participates. For example, Amazon might have a number of sources for a product, number of courier services for home delivery etc. but for an order it will choose a particular courier service and one or more of the available sources of the products. An online shopping transaction in Amazon also requires data write (update) operations in cases such as change of delivery status or shipment status or payment status, etc. For simplicity, we assume that each entity owns a single *data source,* e.g. a denormalized form of a database. Our proposed model offers a framework for secure processing of any general collaborative application which involves data read or write We introduce one independent entity called Service Provider (SP) to facilitate the collaboration. Each partner (entity) has a certain degree of trust on SP, which may change over time. We assume that SP knows the data definition of each entity. Similarly, each entity has to know the identity of SP. CIS is composed of three components namely the adversarial model, security framework and computational model.

2.1 Adversarial Model (AM)

The security assumptions regarding the adversaries are expressed through the privacy constraints desired by a player vis-à-vis his opponents (partners in collaboration and the SP) through a *privacy template* (section 3) built on privacy issues and protections (Pal and Bose 2016).

2.2 Security Framework (SF)

Security framework of CIS is guided by three phases. The initial phase which is the *setup* phase of a specific service or an application chooses the collaborating entities – the data sources (DS) and the SP. This phase is also responsible for constructing the global data dictionary coupled with the global privacy template. Once the setup is done, the system is ready to accept any transaction processing request of a user. This is accomplished by a *transaction specific setup phase* followed by the *transaction execution phase*. The transaction setup phase allows the user to specify its privacy restrictions through local data dictionary and a local privacy template which will supersede the global restrictions. Generally, SP is positioned as the entity to carry out the setup jobs. In the transaction execution phase a sequence of sub processes involving a subset of data sources is constructed. This is a distributed processing scenario which may allow some sub processes to execute in parallel (Pal and Bose 2016).

2.3 Computation Model (CM)

N entities (organizations) with their data sources, DS_1 ... DS_N collaborate to provide an information service to their customers. The service provider SP is chosen by the entities or SP chooses himself. SP has full knowledge of the data definition of each DS. The user or customer C obtains services from this set of data owner entities. C hosted on one such entity sends his request to the transaction related entities either directly or through SP. The query formed to service the request has three parts, *<command, target clause, predicate clause>*. The *command* is *Retrieve, Insert*, *Delete* or *Update*. The target *clause* for retrieval is a set of *expressions* i.e. functions defined on the attributes from DSs. *Delete* command in its target list refers mainly to rows (occasionally columns) in one or more database tables. *Insert* and *Update* have an additional clause representing the values to be put in for a new record or for replacing the contents of an existing record. The predicate clause is a logical expression involving the attributes of different DSs.

SP builds g*lobal data dictionary* by aggregating individual data dictionaries of all DS. Privacy template of each DS specifies its privacy requirement against all other participants C, SP and other DS for any read or write operation. This includes privacy of any data element or attribute of the DS, its identity, schema, data or any part of the results. Similarly, it also includes the privacy of C regarding its identity, the query raised by it and the results or any part result. Note that query or transaction could be an important thing to be protected from being disclosed or misused when it is an isolated or ad hoc query as in the case of IRaaS (Pal and Bose 2013), but may not be so critical when each such query is part of an application requiring joint computation. For write operation the target DSs need protection of intermediate results. These results imply knowledge of the value used for write operations like insert and update. More importantly it can manipulate information wrongly. Other than the DSs both C and SP also specify their privacies required from other participants. In more general case each data source may ask different privacy requirements against different customers or different types of customers. SP constructs *global privacy template* from all these privacy templates. The global template and schema are captured and constructed for all the participants in the system. However, the system may also create local schema and template specific to any query or transaction from the relevant subset of participants which helps to redefine or override privacy constraint expressed by each in a given situation. For example, one DS (say a Bank) may be very cautious against any party but if the requirement of query is from an investigation agency it might behave differently.

During execution a query is broken down into a set of *sub-queries* which are executed in stages to be performed in sequence. A sub-query may have to be further split into a set of *query components* one for each DS. However, the query components within a stage can work in parallel. The decomposition of queries, query execution plan and results are determined based on the local schema of the DSs.

Privacy template describes the privacy constraints imposed by each party w.r.t. other parties, in other words the privacy relationship that should hold between any two parties. Enforcing the privacy constraints will ensure trust in the information service. Cryptographic protocols will be brought in for this enforcement. The privacy and security framework must ensure that SP cannot interfere with data, query or results belonging to the organizations except beyond the point where his accesses are allowed. The sequences of accesses may not be apparent during query decomposition and result reconstruction phases. The actual access patterns may be decided dynamically depending on, which security protocols are available and designed based on the privacy templates. Some examples are:

The most important development *Homomorphic encryption* allows direct operation on encrypted data (Gentry 2009; Olumofin and Goldberg 2010; Shiyuan, Agrawal and Abbadi 2011; Shiyuan, Agrawal and Abbadi 2012). *Private information retrieval* (PIR) helps data anonymization encrypting database index. Indexes are used to quickly access records based on key values (Olumofin and Goldberg 2010; Reardon, Pound and Goldberg 2007). Similarly, *Commutative encryption* is used to exchange information between parties A and B via P who knows the identities of both A and B, unknown to each other, e.g. A could be a customer, B a data source and P the SP (Agrawal, Evfimievsk and Srikant 2003).

3 PRIVACY MODEL

The privacy model in (Pal and Bose 2016) safe guards the privacy concerns of all participants, C, SP and the DSs in CIS to prevent disclosure of sensitive information. CIS has identified seven primary privacy concerns of the participants, namely Identity (I), Schema (S), Data Read (DR), Data Write (DW),

Result (R), Query (Q) and Query Distribution (QD). They are referred as *privacy issues*. The work of Pearson (Pearson, Shen and Mowbray 2009) has also looked into the user selection of privacy preferences from a range of choices. The privacy issues were first introduced in (Pal and Bose 2015).

3.1 Privacy Issues

The privacy issues are meant for different access levels allowed by any party to any remote entity for different kind of operations on the former's data sources. Note access to one's data source is possible only when the identity of that party as well as the data definition (schema) is disclosed through the schema definition. We briefly discuss these privacy issues here.

a. Identity Privacy mainly involves knowledge required to locate and identify a party to communicate with it. To prevent on-line identity discloser anonymous communication should be employed. Off-line identity disclosure can be avoided by adopting cryptographic measures (Pearson, Shen and Mowbray 2009).

b. Schema Privacy refers to the protection of individual schema of the data sources. Cryptographic measures to prevent schema disclosure are not known to exist though it appears to be a serious privacy need in selective cases.

c. Data Read Privacy refers to the protection of data contents (attribute values) of a DS for controlling accesses from other DSs, and C as well as SP. For solving a query often data need to be shared with others. Data obfuscation is one possible cryptographic technique to be applied here. There is a distinction between schema and data privacy. Schema privacy automatically implies data privacy, but the reverse is not true. Even if there is no requirement of confidentiality yet adequate precautions have to be taken to prevent data from being manipulated affecting *availability* and *integrity*.

d. Data Write Privacy refers to an entity protecting its data from being written over by others.

e. Result Privacy refers to protection of the results which are provided to C by SP. As such C can also combine the part results obtained from different entities. C may not like to reveal the final query result as well as part results to any third party.

f. Query Privacy (Transaction Privacy) refers to the protection of the user query from SP and DSs. C is particularly interested to protect the sensitive parts of a query. A transaction text may be disclosed to SP, but the semantics need not be disclosed. Though the textual content of a query may not disclose the complete intention of the user, the constants supplied by the user at runtime are private and must be protected (Olumofin and Goldberg 2010).

g. *Query Distribution (Transaction Distribution) Privacy* refers to the protection of knowledge of query distribution by SP from DSs and C. Note SP distributes the query components to a set of DSs based on their availability and suitability. This protection works at two levels. At the first level there are DSs who are not involved in the query execution and hence need not have any idea about the ongoing query. At the second level DSs who are involved would not be informed about others' participation. This information also may be kept from the client. However in a good collaboration many of these restrictions may not exist.

CIS has addressed seven privacy issues required at the most basic level. For higher granularity of privacy any issue may have to be split generating more issues. Data Write privacy issue may be split into Data Insert, Modify and Delete issues. Similarly, the Schema privacy issue may be split into Schema Read and Write issues, or even give rise to Index privacy issue. Privacy can be defined for each category or role of users. Another point of interest could be that the privacy template is minimized if local schema and even their subsets of relevant attributes are used in place of the entire global schema.

3.2 Privacy Types

A privacy type refers to a particular combination of protections available in a privacy issue. It can be compared to a tuple in a database table. However, there is a fundamental difference here. Unlike databases the protections are not unique or do not require separate existence. For example, protection of customer from service provider is applicable to both identity and query privacy issue. A privacy issue is represented as a matrix, each column represents a *privacy protection* and row represents a *privacy type*. A privacy protection refers to the protection of one party (A) from another party (B) i.e. *A* protected from *B*, or conversely, *A* open to *B* with respect to the underlying privacy issue and hence it has only two possible values "Yes" (*y*) or "No" (*n*). A privacy protection can be compared to an attribute of an entity.

3.2.1 Type and Type-Subset

Let P be a privacy type. The set of privacy

protections in P is denoted by *protection* (*P*). Sometimes privacy types are labeled for easy reference (e.g. Qd privacy). The set of types in P is denoted by type(P). P1 is a type-subset of P if type(P1) ⊆ type(P). Similarly, P2 is a protection-subset of P if protection(P2) ⊆ protection(P).

3.2.2 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. P(c) or P(Q) could be a type-subset, protection-subset or both of P. A privacy issue having m privacy protections can have maximum of 2^m privacy types.

Let us look at the privacy issues at greater details. Consider the issue of identity privacy. It concerns each and every party and it is the gateway for accessing a party. In our model we have N+2 parties – N number of DSs, SP and C. So (N+2)×(N+1) one way communications or access rights are possible among these parties which result in a maximum of $2^{(N+2)(N+1)}$ *privacy types* for identity privacy. If there are M categories of customers who have the same identity privacy requirements, then the total number of privacy options will be $M\times2^{(N+2)(N+1)}$. Our objective is to find out permissible communications between any two parties.

3.3 Enumeration of Privacy Types

For determining the privacy types of different privacy issues there is a common phenomenon: *symmetric* vis-à-vis *non-symmetric*. Privacy relationship, i.e. privacy constraints between any two players A and B is of two types, e.g. A protected from B = 'Yes' or 'No'. Consider p players: B_1 …, B_p and A. Total number of privacy types between players {A and B_1}, …, and {A and B_p} is 2^p, $p\geq2$, i.e. A protected from B_1 = 'Yes' or 'No', …, A protected from B_p = 'Yes' or 'No' etc. Here B represents the players in the same category, e.g. data sources, meaning that their privacy concerns may remain identical. Again the total number of privacy types between any B_i and B_j, i, j=1, ... , p is $2^{p(p-1)}$. Combining this with the relationship of A with Bs the number of privacy types rises to $2^p + 2^{p(p-1)}$. This describes the most general situation, referred to as non-symmetric case. In the symmetric case there is no behavioural difference between any two B_i and B_j w.r.t. A, i.e. the relationship between A and B_1 is same as that between A and B_2. The total number of distinct types in this case reduces drastically to 2. For example, when p = 2 number of types for the non-symmetric case is $2^2 + 2^{2(2-1)} = 8$ but the number of types for the corresponding symmetric cases is 2 + 2 = 4. In the template definition for different privacy issues these will occur. In the remaining discussion Data Privacy Issue has combined *Data Read* and *Data Write*. The summarized version of all the privacy issues and types is depicted in Table 1.

Table 1 shows Identity privacy for Customer and Data Sources (symmetric case). It has three protections, i) C protected from DS, ii) DS protected from C, and iii) DS protected from other DSs. This

Table 1: Summarized version of all the privacy issues (excluding Data Write).

Privacy Issue			**Protections** *a*→ *b*: a protected from b; *a*↔ *b*: *a* protected from *b* and *b* protected from *a;*	**# Privacy Types (2^z)**	
				Symmetric Case (z1)	**Non-symmetric Case (z2)**
Identity Privacy			C ↔ DS, DS ↔ DS	3	N(N+1)
Schema Privacy			DS → C, DS ↔ DS	2	N^2
Data Privacy			DS → C, DS → SP, DS ↔ DS	3	N(N+1)
Result Privacy	Intermediate Result		DS → C, DS → SP, DS ↔ DS	3	N(N+1)
	Final Result		C → SP, C → DS	2	N+1
Query Privacy	Complete Query	Syntax and Semantics	SP → DS	1	N
		Semantics-sensitive constants only	C → SP	1	-
	Part Query	Syntax and Semantics	DS → C, DS ↔ DS	2	N^2
		Semantics-sensitive constants only	DS ↔ DS	1	N(N-1)
Query Distribution Privacy (Identities protected)		Open QD	SP → C: No, SP → DS: No	0	-
		C-Open QD	SP → C: No, SP → DS: Yes		
		DS-Open QD	SP → C: Yes, SP → DS: No		
		Closed QD	SP → C: Yes, SP → DS: Yes		

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 labeled 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 other. Since we haven't put any condition on the issue, there are all 2^3=8 types. If all possible communications are allowed between any two parties – C and DSs there will be n(n+1) protections and $2^{n(n+1)}$ 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. Consider Closed Query Distribution, where the customer and the data sources are unknown to each other, the only privacy type allowed for Identity privacy is (Yes, Yes, Yes) – the Closed/Private type, whereas the Open Query Distribution allows only (No, No, No) – the Open / Public type. Table 2 shows restricted identity privacy which allows identity sharing as guided by Query distribution policy.

Table 2: Identity conditioned by Query distribution.

Privacy Type	Identity Privacy Protection			
	C from DS	DS from C	DS from other DS	Corresponding Qd Type
0 (public)	No	No	No	Open Qd
1	No	No	Yes	C-Open Qd
6	Yes	Yes	No	DS-Open Qd
7 (private)	Yes	Yes	Yes	Closed Qd

4 PRIVACY ALGEBRA

We denote the privacy issues by capital letter and privacy protections by small letters in the following discussions.

Join of Privacy Issues: Let privacy issue X has p protections and Y has q protections of which r are common.Then X.Y represents a new privacy issue obtained by joining X and Y. Join is performed in the same way as database relations are joined. We can also define the joint privacy issue X.Y as X U Y – X ∩ Y. Hence, X.Y will have p + q – r protections. If X and Y are two *independent* privacy issues having no common protections, then X.Y will have p+q protections. For example, join of Identity Privacy of C and DS which has 3 protections C → DS, DS → C and DS → DS (Table 1) and Schema Privacy of Data Source which has 2 protections DS → C and DS → DS (Table 1) results in Identity and Schema Privacy having 3 protections (Table 3).

Table 3: Join of Identity and Schema Privacy of C and DS.

Identity and Schema Privacy Protection of			
C from any DS	Any DS from C	Any DS from any other DS	Privacy Type
*	*	*	0 – 7

4.1 Dominance Relations

Dominance of Privacy Protections: Let x and y be two privacy protections of a privacy issue P. We say x dominates y over P if, in P if x is protected then y is also protected, i.e. protection pair (x,y) cannot assume the value (Yes, No). This is denoted by x > y over P.

Dominance of Privacy Issues over Protections: Let p and q be two protections of privacy issue X and Y respectively. We say p>q if p>q holds in the privacy issue X.Y. We term this as dominance between two privacy issues, i.e. X dominates Y, denoted X > Y over protection p and q. If p and q refer to the same protection, then we say X dominates Y over p denoted by X > Y over p. By applying dominance relations between privacy issues or privacy protections one essentially conditions the joint privacy issue or protections, i.e. obtains conditioned privacy issues

Binary String Representation of Valid Privacy Types: Let a and b be two privacy protections of a privacy issue such that a>b. The set of valid privacy types for (a,b) is {(1,1),(0,1),(0,0)}, i.e. (1,0) is not a valid. In other words for a = "Yes" only value that is allowed for b is "Yes". We can also use binary strings to represent the privacy types in more compact form. The valid types are represented by three binary strings, {11, 01, 00} is equivalent to {11, 0*} or, {1^2, 0*}, * represents either 0 or 1. Therefore, 0^*2^13 represents four strings, namely, 000111, 001111, 010111 and 011111.

Transitivity of Dominance Relations:

Claim 1: *The protection dominance is transitive, i.e. if x > y and y > z then x > z, where x, y and z protections belong to same or different privacy issue.*

Proof: Let x > y and y > *z* over P. The valid privacy types of both (*x*, *y*) and (*y*, *z*) are {11, 01, 00}. By joining these on common values of y we get {11, 01, 00} as the set of valid privacy types of (*x*, *z*), i.e. x > z over P. Note, this logic holds even if *x*, *y* and *z*

belong to different privacy issues.

Claim 2: *The privacy issue dominance is a transitive relation, i.e. X > Y and Y > Z then X > Z, where X, Y and Z are privacy issues over a common privacy protection or over different protections*.

Proof: Let p, q and r be three protections of privacy issues X, Y and Z respectively. X > Y => p > q in X.Y and Y > Z => q > r in Y.Z. By Claim 1, Y > Z => p > r in Y.Z. Again p and r are protections of X.Z. By definition of privacy issue dominance we can conclude X > Z.

Valid Privacy Types: Let $b_1, \ldots, b_k$ $k \geq 2$, are k privacy issues w.r.t. to a common privacy protection. Alternatively, let b_1, ..., b_k be the k privacy protections over the domain of one or more privacy issues (which have been joined). For the joint protection domain (b_1, ..., b_k) total number of possible privacy type is 2^k. We can represent these types by binary strings of length k, i.e. k-bit strings whose values range from 0 to $2^k - 1$.

Claim 3: *Let $b_1 > \ldots > b_k$ hold for the joint protection domain (b_1, ..., b_k). Then, there are only k + 1 valid privacy types $\{1^k, 0^1 1^{k-1}, \ldots, 0^{k-1}1^1, 0^k\}$, equivalently $\{2^k - 1, 2^{k-1} - 1, \ldots, 2^0 - 1\}$ out of a total possible 2^k types.*

Proof: We prove the claim by induction. If $b_1 > b_2$ the valid types are {11, 01, 00} or, $\{1^2, 01, 0^2\}$. This proves that the claim holds for k=2. Let the claim hold for k = i. We will show that it holds for k = i+1. The case k = i indicates that $b_1 > \ldots > b_j$. Assume, $b_i > b_{i+1}$. So, in the combined privacy (b_1, ..., b_i, b_{i+1}) obtained by joining (b_1, ..., b_i) and (b_i, b_{i+1}), we have $b_1 > \ldots . b_i > b_{i+1}$. For k = i, we get i+1 valid privacy binary strings. For k = i + 1, only one bit will be introduced in the right hand side of each string. We call this bit parity bit. Thus, there are i+1 strings with parity bit 0 and i+1 strings with parity bit 1. Since, $b_i > b_{i+1}$ we can discard the strings having parity bit 1 except the strings of all 1s. Thus, the number of valid privacy strings will be i + 1 number of strings having parity bit 0 plus the string with all 1s. Clearly, the combined privacy has i + 2 privacy types $\{2^{i+1} - 1, 2^i - 1, \ldots, 2^1 - 1, 2^0 - 1\}$. This completes the proof.

5 PRIVACY ALGEBRA AND CIS

In this section we demonstrate how privacy algebra can be used to simplify and consolidate the privacy issues of CIS. First we notice that successive applications of join of elementary privacy issues are not affected by the sequence in which the operands are selected for the join operation. For example, A.(B.C) = (A.C).B = C.(A.B) = B.(C.A). We will detect the interdependencies in the form of dominance relations between privacy issues over privacy protections or vice versa. Let us allow all possible communications between any two parties out of n+2 parties, C, SP, DS_1, ..., DS_n, i.e. no symmetry among DSs are assumed. We consider five privacy issues: I (identity), S (schema), D (data), Q (query) and R (result). The query distribution (Qd) issue is fixed at (No, No), i.e. *Open-Qd* issue (Table 2). For each of the five privacy issues there are (n+2)(n+1) one way accesses. Therefore, total number of possible privacy types for all five issues is $2^{5(n+2)(n+1)}$. With increasing number of participants or large value of n it may not be practical to implement so many privacy types. We therefore look into reduction of allowable privacy types considering dominance relations of different privacies (Section 4).

Let us consider the privacy protection between any two DS across different privacy issues. Note, for this protection, the following dominance holds: I > S > D, I > R and I > S > Q because without access to identity one cannot have access to schema. In other words, when identity is protected schema cannot remain unprotected. But when identity is unprotected schema can be protected or unprotected. Similarly, without learning the respective schema learning data would not be possible, but for learning the result part the knowledge of schema need not be essential. 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 {11, 01, 00}. Similarly, for S > D we also have 3 valid privacy types {11, 01, 00}. By Claim 3, joining I > S with S > D we have 4 valid privacy types $\{111, 011, 001, 000\} = \{2^3 - 1, 2^2 - 1, 2^1 - 1, 2^0 - 1\}$ for the dominance I > S > D shown in Table 4. Similarly, for I > S > Q, we also have 4 valid privacy types {111, 011, 001, 000}. By joining privacy types of I > S > D with those of I > S > Q the set of valid privacy types for I, S, Q, and D are shown in Table 5. Joining I > R having 3 privacy types {11, 01, 00} with (I>S>D).(I>S>Q) with 11 types (Table 6).

Thus by applying the dominance relations we have been able to reduce the number of privacy options between two data sources from $2^5 = 32$ to 11. Similarly, considering the protection of DS from C, given relations I > S > D and I > R we get 7 privacy types (Table 7). Protection C from a DS involves only identity privacy, and has two types: "*". Protection DS from SP involves data and result privacies which are independent of each other, hence

all four types (*,*) are valid.

Table 4: Protection of DS from DS: I > S > D.

I	S	D	Type	#types
No	No	No	000	1
No	No	Yes	001	1
*	Yes	Yes	*11	2

Table 5: Joint Protection of DS from DS I>S>D & I>S>Q.

I	S	D	Q	Type	#types
No	No	*	*	00**	4
*	Yes	Yes	Yes	*111	2

Table 6: Protection of DS from DS for the join (I>S>D).(I>S>Q).(I>R).

I	S	D	Q	R	Type	#types
No	No	*	*	*	00***	8
No	Yes	Yes	Yes	No	01110	1
*	Yes	Yes	Yes	Yes	*1111	2

Table 7: Protection of DS from C for I > S > D and I > R.

I	S	D	R	Type	# of types
No	No	*	*	00**	4
No	Yes	Yes	*	011*	2
Yes	Yes	Yes	Yes	1111	1

5 CONCLUSIONS

Pal and Bose (2013; 2015) proposed IRaaS, a user defined security model for the information retrieval services from different data sources for customers with help of a service provider. CIS (Pal and Bose, 2016) has added *write* operation beyond *read* and also considered the problem as a collaborative model for any arbitrary joint information exchange activities. It extends CIS privacy model further by introducing the concept of *privacy algebra* which incorporates privacy constraints of different partners. It has used the idea of dominance between the privacy issues and obtained results for concise expressions of the user privacy requirement, a more compact privacy template. Further depth in privacy template can be achieved by restricting accesses to attributes rather than relations. Optimizing the privacy template by relaxing or restricting some constraints is another point of study. Developing general collaborative applications along the same line while taking care of user or role concern for the security of the owner and remote sites and a language development (high level and low level) with a compilation process needs to be looked at. As next generation systems will be highly collaborative and will have to share information based on a planned privacy model, interoperability via open communication and standardized data exchange is needed (Karnouskos et al. 2012).

REFERENCES

Agrawal, R., Evfimievski, A. and Srikant, R. (2003). Information sharing across private databases. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*.

Ahmed, T. and Tripathi, A. (2007). Specification and verification of security requirements in a programming model for decentralized CSCW systems. *ACM Transactions on Information and System Security (TISSEC)* 10.2 (2007): 7.

Karnouskos, S., Colombo, A., Bangemann, T., Manninen, K., Camp, R., Tilly, M., Stluka, P., Jammes, F., Delsing, J. and Eliasson, J. (2012). A SOA-based architecture for empowering future collaborative cloud-based industrial automation. *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*.

Olumofin, F. and Goldberg, I. (2010). Privacy-preserving queries over relational databases. *Proceedings of the 10th international conference on Privacy enhancing technologies*. Springer Berlin Heidelberg, 2010.

Pal, A. and Bose, S. (2013). Information Retrieval as a Service for Multiple Heterogeneous Data-Privacy Model. *Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, Stirlingshire, Paper 31.

Pearson, S., Shen, Y. and Mowbray, M. (2009). A Privacy Manager for Cloud Computing. *Lecture Notes in Computer Science*, *Cloud Computing*. Springer, 2009, pp.90-106.

Zhang, X., Nakae, M., Covington, M. J., & Sandhu, R. (2006, June). A usage-based authorization framework for collaborative computing systems. In *Proceedings of the eleventh ACM symposium on Access control models and technologies* (pp. 180-189). ACM.

Reardon, J., Pound, J. and Goldberg, I. (2007). Relational-complete private information retrieval. *University of Waterloo, Tech. Rep. CACR* 34 (2007).

Shiyuan, W., Agrawal, D. and Abbadi, A. (2011). Towards practical private processing of database queries over public data with homomorphic encryption, *Technical Report* 2011-06, *Univ. California at Santa Barbara*, 2011.

Shiyuan, W., Agrawal, D. and Abbadi, A. (2012). Is homomorphic encryption the holy grail for database queries on encrypted data. *Technical report, Department of Computer Science*, UCSB, 2012.

Gentry, C. (2009) A fully homomorphic encryption scheme, *Diss. Stanford University*, 2009.

Pal, A. and Bose, S. (2015). Information Retrieval as a Service - IRaaS: A Concept Paper on Privacy

Analysis, WPS 763, Indian Institute Management Calcutta, June 2015. https://facultylive.iimcal.ac.in/sites/facultylive.iimcal.ac.in/files/WPS%20763.pdf.

Pal, A. and Dey, S. (2014). A Policy-based Security Framework for Storage and Computation on Enterprise Data in the Cloud, Proc. 47th Hawaii International Conference on System Sciences 2014 (HICSS), 6-9 January 2014, Hawaii. IEEE.

Pal, A. and Bose, S. (2016). Collaborative Information Service: The Security Question, Proc. 49th Hawaii International Conference on System Sciences 2016 (HICSS), January 2016, Hawaii. IEEE (to be appeared).

Atallah, M. (2006). Security issues in collaborative computing." *Lecture Notes in Computer Science* 4112 (2006): 2.

Zhu, M., Shen, J., Yan, S., & Zhao, B. (2006). *U.S. Patent No. 7,069,298*. Washington, DC: U.S. Patent and Trademark Office.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS