Collaborative Information Service: The Security Question

Asim Kumar Pal
Indian Institute of Management Calcutta, Kolkata, India
asim@iimcal.ac.in

Subrata Bose
Neotia Institute of Technology, Management and Science, Kolkata, India
subratabose@yahoo.com

Abstract

Growth of computing and analytics, internet and distributed computing, and increasing tendency to use crowd sourcing for creative tasks, the use of collaborative idea in information intensive activities is a growing research concern. In CIS companies would cooperate to do their interlinked business together. It responds to customer requests for information retrieval from or writing on data contents. We study the role of a flexible privacy model that can simplify the job of maintaining security in a collaborative environment. There has not been sufficient effort put in this direction though vast amount of literature exists on security in databases, distributed databases and cloud. Not many attempts for the security of collaborative information retrieval systems either exist. IRaaS – Information Retrieval as a Service – introduced 'privacy template', whereas the explicit expression of privacy constraints adopted by CIS for security implementation. The rational approach to security for such problems is a novel idea.

1. Introduction

Let us look at the following information activities:
(1) Show me houses from different builders along with prices and payment schedule, landscape, apartment plan and interior design within a budget range and having a few location features.
(2) Find a criminal along with records of his criminal activities from different police stations in the city given a few voice samples.
(3) Locate a terrorist having multiple identities (aliases) on the move.
(4) Strike deals online with reputed designers and dress makers for making costumes for a dance troupe going on a world tour.
(5) Stop fraudulent bank transactions across the globe connected with a group of suspected criminals.

The information processing activities have the following features (the numbers in the parenthesis refer to the above mentioned activities):

a. Collaboration among different data owners is needed.
b. All data sources needed may not be known a priori, some may become apparent only during execution (3,4).
c. Data sources can belong to a single (1,2), or multiple domains, (3-5). The requirement of multiple domains may arise from differences in types of data (4), systems (5), or both (3) [1,22].
d. Information activities could be retrieval (read) (1-3) or transactional (write) operations (4,5).
e. Maintaining privacy of each player, a data provider or a seeker of information, is crucial to the success of an activity.
f. Authorized people can only perform 'sensitive' activities (2,3,5). The players need proper access rights for both read and write activities.
g. Query, data or result could be in fixed format (structured) or in variable format (unstructured) such as for text, picture, audio or video data.

We envisage a number of organizations who want to do business with their data. We are interested in developing a collaborative system to respond to information requests from their clients. A client wants to interact with the organizations for receiving information of his interest. The client can also be allowed to modify the content (data values) on one or more data sources. The parties want to collaborate because their activities or data are linked with each other.

No need to say that data is exploding. There are many companies that own data will like to get the value out of it. They come together and create a collaborative information service (CIS). Different portals are such examples. The online stores like Amazon and Flipkart are such examples who connect different manufactures, vendors and courier services. Similarly travel and hospitality industry have reasons to come together. Police stations within a city, states within a nation, and even nations under some arrangement can work together to fight against criminal and terrorist activities. Airlines form networks. One Airline can belong to different networks [1,22]. Different companies in a supply chain may
collaborate. There are large multinational companies who have the requirement of information processing from any of their locations. For such organizations whose data sizes are huge, dynamic and complex, a central control may not be feasible rather a distributed processing mechanism would be more useful. Their different departments along with other partners like vendors, etc. can be considered to be separate organizations having related activities.

All these are examples where collaboration is necessary among homogeneous or heterogeneous data sources but protection is desired from each other because of strategic interests. The data may be spread over a number of databases some of which may be on clouds. One may argue that with the presence of cloud computing the security question does not arise. But to make the cloud trustworthy, one or more clouds can collaborate such that the partners can work freely among themselves [12,14,30]. Similarly a big cloud may treat its different sites as independent organizations that can collaborate among themselves to provide more security [31]. In such cloud a CIS can exist. For cooperative effort the best examples are the crowd sourcing tasks. Cooperative planning [23,24] and cooperative method of developing trusts in maintaining an Enterprise System on the cloud [25,32] are other examples.

The authors proposed a similar system called Information Retrieval as a Service (IRaaS) [1,22] and discussed its taxonomy (Open, Closed, Collaborative IRaaS and Hybrids). Whereas CIS proposed here has a much broader mandate to support a collaborative effort that is required from data organizations for doing their interlinked businesses rather than just allowing a customer to seek information from different data sources in IRaaS. Note that, Collaborative IRaaS [1] is a loose collaboration between different IRaaSSs, whereas CIS is designed for a stronger collaboration between different organizations.

Our task is to develop a security framework for incorporating different types of privacy requirements of the players. Specific cryptographic protocols needed for execution will be built on this framework. Google and Wikipedia are two best examples of information service, yet these are not CISs because they handle static data and do not perform any computation.

This work tries to differentiate privacy from security in the context of a collaborative information service though there is a common belief that the two terms are almost identical. Privacy is the basic right of an individual or an organization to protect its identity and other possessions. It is the right to be let alone, or freedom from interference or intrusion. Security is the tool to maintain privacy. This work aims to provide a security framework in which each individual player is allowed to express explicitly its privacy needs vis-à-vis other partners in a collaborative scenario.

The importance of collaboration cannot be ignored in the days of increased use of computing and analytics, growth of Internet and distributed systems and the increasing tendency to use crowd sourcing for creative tasks. Thus the use of collaborative idea of information activities while maintaining privacy is a growing research concern. We study the role of a flexible privacy model that simplifies the task of maintaining security in a collaborative environment. It is viewed that effort put so far in this direction is inadequate, though there is vast amount of literature on security in databases, distributed databases and cloud. Existing works in collaborative information retrieval systems are not known to focus on the security issues. Use of an explicit and comprehensive privacy model for developing a security system of an information service has not been sufficiently researched. [1,22] is an exception where a detailed privacy model was proposed for the first time.

1.1. Contributions

A comprehensive security model for the collaborative information service has been suggested. CIS has also been designed to take care of write operations (beyond read) – this is a novel feature. The computational model and the security framework have been spelt out which explicitly incorporates privacy constraints taken from different partners. This would be a novel phenomenon. The idea of privacy template and its constituents have been introduced. The service of a third party to mediate between the partners has been conceived to implement security. Other novel features would be “Rational” approaches for solving the security for such a practical problem, and isolation of “security” and “privacy” concepts.

2. Related Work

The section describes areas of research those are relevant and closely related to CIS.

Privacy Preserving Query Processing (PPQP) -

Over the years, research community has developed a wide range of PPQP techniques [3,4,5]. These techniques protected data and result privacy but query privacy has not been given considerations. The work of Hildenbrand et al. [9] proposes an encryption scheme to keep encrypted data in cloud and process a range of SQL queries on the encrypted data. It is a building block of Database as a Service (DaaS). Homomorphic encryption is a common technique for preserving data privacy and query privacy in PPQP [6,13,15].
encryption by Craig Gentry [16] allows computing arbitrary function on encrypted data but unfortunately it is still too slow for large-scale data processing [9] and its practical implementation is yet to be established. Wang et al. [15] showed that a fully homomorphic encryption scheme is capable of processing a complex selection, range, join or aggregation query on encrypted data. The authors did not work on the practicality issue of fully homomorphic encryption. In [7, 8] the authors have shown how private information retrieval (PIR) helps PPQP by hiding sensitive constants of clients query.

Mediation Service for Data Integration from Heterogeneous Sources – Providing IR service from the data owned by different independent and autonomous data sources demands integration of heterogeneous data lying in multiple servers [1]. A number of approaches have been proposed in the literature, mediator based approach being the most prominent [17]. In a mediator system user queries made on a single schema is reformulated into queries on the local schema of the respective data sources containing the actual data. The global schema provides a reconciled, integrated, and virtual view of the underlying sources. A central data integration system [2,17] has a global schema and thus provides the customer with a uniform interface to access information stored in the data sources. The Distributed Interoperable Object Model (DIOM) [2] offers a query mediation framework through an adaptive approach to interoperability instead of an integrated global schema.

Cloud Databases – A database that runs on cloud and delivered to user on demand via internet is termed as cloud database such as Amazon EC2. Numerous works both in industry and academia [10] makes cloud database an important natural phenomenon in cloud area. An important leap in this research direction is Big data which describes a massive volume of both structured and unstructured data that is so large that it's difficult to process using traditional database software techniques. CIS needs to operate on both structured and unstructured data, handle massive volume of data and could be hosted in cloud.

Collaborative Computing in Cloud - CIS described in Section 3 requires collaboration among different data sources for data exchange. Query executions in a collaborative cloud [12] 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. Yoon et al. [14] presented a mathematical model for dynamic collaboration of cloud service providers for auction market to offer collaborative services to its customers.

Cloud Services Brokerage - The term cloud service brokerage (CSB) was coined by Gartner [19]. CSB refers to third parties or intermediary whose job is to negotiate the relationships between the cloud service provider and its consumer, integrating various services to meet a user’s cloud computing need. CSB acts as an agent between an enterprise and cloud service providers. A cloud user may have a need that cannot be fulfilled by any single cloud service provider but requires aggregation of several service providers [18]. The role of the service provider in CIS system has distinct similarities with that of a CSB.

Privacy in cloud computing - Pearson [21] identifies number of privacy sensitive information which includes personally identifiable information (PII) such as name, address, credit card number, postal code, Internet Protocol (IP) address, sensitive information such as personal financial information and job performance information etc. Identification of private information is the primary task of privacy protection [11]. Pearson et al. [20] described a privacy manager for cloud computing. The privacy manager helps users communicate their choices for the use of their personal data to the service provider. A user might want to be anonymous, and might wish partial or full disclosure of his identity. The Privacy Manager allows the user to express privacy preferences about the treatment of their personal information, including the degree and type of obfuscation used [20].

Collaborative Information Retrieval - Information retrieval in collaborative manner has been studied in literature. The authors [26] have investigated the problem of collaboration in the context of handling of patent applications. In [27] tourists' collaborative information search behavior in detail, including their search stages, online search strategies, and information flow breakdowns has been studied. A model of tourist collaborative information retrieval was developed. In [28] the authors use role mining to optimize the collaboration within a search session. The authors in [29] have proposed a four-dimensional model for collaborative search that can be used to characterize existing systems. A four layer model composed of information – tools – user – results for collaborative information seeking has been proposed in [33].

3. Collaborative Information Service (CIS)

CIS to become a powerful mechanism for collaboration among intending partners needs to support the following functions:

a) A number of organizations decide to collaborate to perform their interlinked operations, particularly
providing information activity services to their clients or customers for retrieving from their data sources or for modifying their own data sources based on interactions with the customer. Any given activity would involve one or more of the collaborating organizations. The data sources can belong to the same domain or different domains. The domains differ mainly due to, for examples, different types of organizations, organizations belonging to different states or country which follow their own rule, different departments and stakeholders of a large company or different partners in a supply chain. Same domain implies different organizations having similar data or databases, e.g. police stations within a city. Note every organization has its own data source or database. But often the terms ‘data source’, ‘data organization’ or ‘organization’ have been used interchangeably.

b) The customer need not always have adequate knowledge regarding the contents of the data sources required to formulate the query appropriately. A proper interface between the customer on one side and the organizations on the other would make CIS more flexible and easier to use. Note in general a transaction has been used to indicate a retrieval request or a writing request. Often we have used the terms ‘query’ and ‘transactions’ interchangeably. Unless we are specifically referring to a reading query, we would be referring to a transaction query in general.

c) Data sources may have fixed formats for performing non-multimedia operations or could have more complex formats for supporting text and multimedia operations.

d) An information activity could just be retrieval operations where the customer is interested to obtain answers to his query. In other situations it could be a request from the customer for a write operation on one or more data sources that agree to the request. Note further that a write may indicate in general insert, modify and delete operations.

e) All the organizations are autonomous and independent of each other. Any organization decides access rights to its customers as well as other organizations both for read and write operations on its own data sources.

To achieve above functionalities CIS is proposed to comprise of three components namely the adversarial model, security framework and computational model.

3.1. Adversarial Model

The security assumptions on the adversaries are expressed through the privacy constraints desired by any player vis-à-vis his opponents (all the remaining players) using the concept of privacy template, which in turn uses the concept of privacy issues and protections (Section 4). Here a Service Provider SP has been considered along with other players as organizations (data sources) and the customer. The role of SP in this context could be made more explicit by considering the constraints as external constraints and internal constraints. The former applies to each party vis-à-vis other parties (except SP) and the latter is for SP vis-à-vis all other parties. This adversarial model could be termed as Rational Adversaries as in [25,34].

3.2. Security Framework (SF)

Global Setup: N independent organizations each having its own data source (DS) decide to cooperate for a joint business. Each DS has Data Dictionary and Schema based on its exposable view. The data sources together construct the Global Data Dictionary (GDD). They decide on categories of their customers who need similar treatment as far as security is concerned, i.e. their privacy constraints are similar. The data sources also construct the Global Privacy Template (GPT) to take care of the privacy concerns among them and vis-à-vis the customer for each category. Strictly speaking GDD will incorporate GPT within itself. They choose a Service Provider (SP) as an intermediary between themselves and their customers during the execution of ‘information activity request’ (e.g. a query or transaction). SP could also help in developing the GDD and GPT.

Transaction Setup: For a given transaction q from a customer of some category, n ≤ N parties need to participate. Consequently a local data dictionary (lkd) and local privacy template (lpt) are constructed. In addition the lpt may also have to incorporate additional elements of privacy specific to any individual customer (which is over and above the general privacy restrictions imposed by the partners vis-à-vis the category to which the customer belongs). Note that, when the parties are chosen to execute q (dynamically) during transaction execution lkd and lpt are set to GDD and GPT respectively. SP will use an interface to build the query by interacting with the customer.

Transaction Execution: q is broken down to a series of sub-queries which need to be executed in sequence. Each sub-query is divided further into query components, each of which involves one or more data sources. To implement the security as per lpt the execution phase will apply available cryptographic primitives on the concerned parties for executing query components, sub-queries and the full query, and assimilating sub-results at each stage. SP will provide the necessary intermediation.

3.3. Computation Model (CM)
In this section we present the computational model for CIS. The authors in [1] discussed these issues for IRaaS which is preferably a cloud based service. The security protocol suggested for IRaaS was quite specific, whereas our current security framework is much broader. Moreover the basic privacy issues were brought in there, whereas a broader and in-depth outlook so far as privacy templates are concerned has been considered here. [22] has not looked into the computation model. Both papers have emphasis on different types of information services and taxonomy of information services, whereas we are considering a general service here.

Let us assume that a set of \( N \) organizations with their data sources, \( DS = \{DS_1, DS_2, ..., DS_N\} \) are interested to collaborate to provide a Collaborative Information Service (CIS) to their customers. The \( DSs \) (meaning the organizations representing these data sources) will choose an intermediary, may be on payment basis – rental or pay per use, called Service Provider (SP) who will help them to collaborate. With proper security mechanisms in place it is quite possible to elect SP from the \( DSs \). Even a distributed or collective leadership of \( DSs \) can act as a SP. Conversely an organization who wishes to provide an information service will find out the relevant \( DSs \) who are interested in collaborating for such a task. The former will be known as SP. What we discuss below will not be affected by how SP arrives in the scenario. How the SP will be found out, or elected, or created by a mechanism by \( DSs \) or vice versa is beyond the scope of this work.

Let us assume that SP has full knowledge of the content for each \( DS \), i.e. the data elements (attributes): their names, formats and synonyms, if any; the schema, e.g. data dictionary in a database contains such information. The customer (C) obtains services from the data owners \( DSs \). C sends his request to concerned \( DSs \) via SP. There can be a query interface which helps C to build his query interactively with SP.

Without loss of generality we assume the most elementary syntax for our data sources the relational model, where each data source consists of one or more relations or tables. Each column in a relational table represents a data element called attribute, whereas each row represents a record (or occurrence) consisting of number of values one for each attribute. We use a form of relational calculus, SQL like languages: which has three parts, \(<\text{command}, \text{target clause}, \text{predicate clause}>\).

The command is Select for retrieval purpose and Insert, Delete or Update for writing purposes. The target clause in case of Select command consists of a set of expressions where each expression is a function \( f \) defined on attributes from \( DSs \). Delete command in its target list refers mainly to rows (occasionally columns) in one or more relational 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, which is true or false, involving the attributes of different \( DSs \). The predicate clause can be designed to include operations like grouping, sorting, aggregating or repeating. We focus mainly on Select. The security issues of other write commands are similar to that of Select, except that it may involve one or more \( DSs \) besides C. This is because the privacy issues of all the commands are similar. We are ignoring operations like Merge etc.

As mentioned earlier, the \( DSs \) share their (database) schema with SP for any CIS. It is entirely possible that a \( DS \) is interested only in a set of attributes for activities of one CIS and another set of attributes for a different CIS. Without loss of generality it may be assumed that the exposable portion of the schema, view, of a \( DS \) is a ‘de-normalized relation’. Let the exposed schema for \( DS_k \) be denoted by \( S_k, k = 1, ..., N \). The SP will build the so called Global Data Dictionary (GDD) by aggregating individual data dictionaries of all \( N \) number of \( DSs \).

The privacy template \( P_k \) of \( DS_k \) expresses its privacy requirement vis-à-vis other players \( C \), SP and other \( DS_j, j \neq k \) for any activity – retrieval or writing. The privacy concerns any attribute of \( DS_k \), its identity, schema, data or any part of the results that discloses something about the \( DS \). Similarly it takes care of the privacy concerns of \( C \) regarding its identity, the query and the results or result parts. For a writing operation the target \( DSs \) want protection of any intermediate result which implies certain knowledge about the value used for the write operations like Insert or Update. Both parties \( C \) and SP will specify the protections \( P_C \) and \( P_{SP} \) required from other parties. Note in the most general case each data source may ask for different privacy requirements against different customers or at least different types of customers. SP will construct the Global Privacy Template (GPT) from the privacy templates. The GPT would be incorporated within GDD.

Semantically, the global schema \( S = U\{S_1, ..., S_N\} \) and the global privacy template \( GPT = U \{P_{C_1}, ..., P_{C_P}, P_{SP}, P_{SP}, ..., P_{SN}\} \) with DS \( = \{DS_1, ..., DS_N\} \) for \( N \) data sources and \( C_{all} = \{C_1, ..., C_t\} \) for \( t \) different types or categories of customers. The global data dictionary \( GDD = < DS, S, GPT, C_{all} > \). Similarly, without loss of generality the local schema \( s = U\{s_1, ..., s_N\} \) and the local privacy template \( lpt = U \{P_{C_1}, P_{SP}, P_{SP}, ..., P_{SN}\}, j=1, ..., t \) for DS \( = \{ds_1, ..., ds_N\} \) \( n \) local data sources used in the current operation. And the local data dictionary \( ldd = U\{ds, s, lpt, C_j\} \). Note that, the local objects
applicable to the current request, are subsets of corresponding global objects: \( ds \subseteq DS, s \subseteq S, p \subseteq P, C \in C_{all} \) and \( ldd \subseteq GDD \). \( GDD \) and \( GPT \) remain static as long as no new \( DS \) joins the CIS or an existing \( DS \) makes any change to its list of attributes or their privacy concerns, etc., whereas the life of an \( ldd \) is valid only as long as the transaction is live. In a complex situation the \( dd \) along with its components may not be known fully before the computation starts. It will be discovered later dynamically during the execution phase.

We now discuss what is known as transaction construction phase. Note a query need not concern all the attributes for the participating data sources. Let \( q \) be the query built by \( SP \) by coordinating with all the partners. Note we are making a distinction between an informal question and a (structured) query. In practice a customer can start asking the question informally, which would be gradually converted to \( q \) with the help from \( SP \) who uses \( GDD \). This process of conversion could be a long one and is also fraught with errors. This will depend on the quality of the schemas, particularly the synonyms or aliases used against an attribute for a given \( DS \). It is also desirable to have more than one format for an attribute to allow flexibility in the possible values. Note \( q \) will be defined on the local schema \( s \).

Let us now come to result phase or transaction execution phase. Query \( q \) subsequently is broken down into a set of \( j \) sub-queries \( q^1, \ldots, q^j, j \geq 1 \), which will be executed in \( j \) stages. Note \( q^k \) may require part-results \( r^k \) obtained from \( q^1 \) along with data from some or all the participating data sources from \( ds_k \), and \( q^3 \) may require previous results \( r^1 \) and \( r^2 \) obtained from \( q^1 \) and \( q^2 \) respectively and data from \( ds \) and so on. In each stage the corresponding sub-query will further be broken down into a set of query components, one for each \( DS \). Let \( q^1_k \) represent the query component (an elementary query) meant for \( ds_k, k = 1, \ldots, n \), against the sub-query \( q^1, l = 1, \ldots, j \). Even if the sub-query \( q^1 \) can be executed only in sequence, the query components within a stage can work in parallel. Let \( r^k_l \) denote the result obtained by executing \( q^1_k \), where \( r^k_l = f(q^1_k, r^1_l, \ldots, r^{l-1}_l, d_2, \ldots, d_n) \), where \( d_2 \) denotes the data for \( ds_k \) and \( f \) is the combination function. Who does this computation for \( r^k_l \)? It is obviously done by \( ds_k \) who is the only party entitled to do this job. Note \( r^1_l = g(r^1_1, \ldots, r^1_n) \) is the result obtained in stage \( l \), with combination operator \( g \). The final result \( r = g(r^1, \ldots, r^j) \). Note the decomposition of queries, query executions and results are determined based on the local schema. Here we have discussed only w.r.t. the retrieval operation.

One thing not explicitly mentioned above is how \( SP \) and the customer \( C \) is engaged during query execution phase. Actually \( SP \) who builds the query based on input from \( C \), is also in a position to decompose the query into sub-queries and query components. Recall \( SP \) only builds the local data dictionary from the global one. It is the responsibility of \( SP \) to distribute the query components among the participating data sources and to construct the intermediate results at every stage by combining the results given by the data sources. Further for more complex queries where all the sub-queries as well as the query components cannot be determined before the query execution starts, these will be determined as the execution proceeds. Similarly, as stated earlier, in a further complex scenario, even the participating data sources may not be fixed a priori, they will be selected dynamically during execution.

So far it seems that role of \( C \) is relevant only during construction of the query and obtaining the result. This could often be the case. But this may not be the case when we consider the privacy issue which will be apparent later. Similarly we have not talked about the role of the privacy template. Coming to the security issue the first thing that is relevant is the privacy template which 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 appearing in a privacy template will ensure trust in the information service CIS. At this stage the cryptographic protocols will be brought in for this enforcement. The privacy and security framework should be such that \( SP \) cannot interfere with data, query or results belonging to the organizations and the customer.

As such the sequences of accesses may not be apparent during query decomposition and result combination (reconstruction) phases. The actual access patterns may be decided dynamically or even statically depending on how security protocols are designed based the privacy templates. A few options suggested at this point are:

a) Homomorphic encryption, which allows direct operation on encrypted data. This is the most powerful method, but so far only a limited number of operations are permitted [6, 13, 15, 16].

b) Private information retrieval (PIR) permits to anonymize the data in the databases using encryption on the indexes. Indexes are used to quickly access the records based on key values [7, 8].

c) Commutative encryption is used to exchange information between party \( A \) to party \( B \) via party \( P \) who knows the identities of both \( A \) and \( B \) but \( A \) and \( B \) do not know each other. For example, \( A \) could be a customer, \( B \) a data source and \( P \) the \( SP \) [3].
Here SP provides the central role in query building and execution by interacting with the customer and the data sources. However privacy requirements may not even allow the query parts to be disclosed in full to SP or to data sources. Even access to the part meant for a particular data source may be restricted to latter. Similarly C may be stopped for checking intermediate results. In such computations between two players A and B a third player C (conduit) is used. Here mostly SP takes the role of the third player C.

Let us look at bit more closely how a query decomposition help in maintaining security. First note that a query q has two parts, syntax and semantics. The knowledge of constants used in target expressions or predicates (including operators) appearing in a query q may be crucial to understand the implications and objectives of the query and for finally obtaining the result. Let us encode these two pieces of information as syntax(q) and semantics(q) respectively. The syntax(q) needs to be disclosed to SP, whereas semantics(q) need not be disclosed. Note that this is one of the major assumptions that syntax needs to be disclosed. Similarly, for the sub-queries and query components these two can be separated. Thus we can state the following result:

Theorem: In CIS CM implements SF based on AM. The accuracy, efficiency, reliability and trustworthiness of execution will depend on i) the efficiency of transaction (query) building by taking input from the customer and partner organizations, ii) efficiency of query decomposition based on cryptographic primitives, iii) privacy requirements among all the players including SP, iv) efficiency of the privacy algebra to construct the privacy templates, and finally v) availability of the security primitives. This is a “rational” approach to security for a real life problem [25,34].

4. CIS – The Privacy Model: Privacy Templates

Unauthorized disclosure of sensitive or private information is a serious concern for both customer (client) and the data sources. The privacy concerns of C are the query and its result; those of DSs are the data and part result. For the service provider (SP) the issue is of winning the trust of both information seeker and givers. Its privacy concern is regarding the query distribution pattern among different DSs and query parts sent to individual DSs for processing. Here C is an unfamiliar entity either for the DSs or for SP, or both. Similarly DSs could also be unfamiliar entities to C, sometimes often unknown or hardly known even to the fellow DSs. Thus identity becomes a privacy issue. Further the strategic or business interests of any DS could conflict with those of other DSs. But quite often the issue of efficiency and cost overshadows that of security. Depending on the complexity of a query it may be beneficial to share the schema, data, result parts or even query parts to a certain extent. To understand the complexity of the privacy problem fully, one has to find out all possible channels of leak. More importantly one should also look into the motivations of other parties to sneak into the data. Similarly each party involved in the computation has to understand the full implications on the efficiency and capability of query execution, both in range and depth. In this work we are interested in developing a privacy model by taking into consideration the privacy concerns of all the players involved. The work of Pearson [20,21] has close similarity in terms of offering the user selection of privacy preferences from a range of choices. The privacy issues were first introduced in [1].

4.1. Privacy Issues

In our model we have identified seven privacy issues which cover all the aspects involved, namely Identity (I), Schema (S), Data Read (DR), Data Write (DW), Result (R), Query (Q) and Query Distribution (QD). We briefly discuss these privacy issues here.

a. Identity Privacy mainly involves knowledge required to locate and identify a party to communicate with it. Disclosure of personally identifiable information (PII) is the identity privacy issue. While IP address reveals online identity, name of the person or organization, address, pin code etc. are offline identities. To prevent online identity discloser anonymous communication should be employed. Offline identity disclosure can be avoided by adopting cryptographic measures [21].

b. Schema Privacy refers to the protection of individual schema of the data sources. For query execution the SP has to prepare a query plan similar in line with that in a distributed database system. Schema here means the global schema S that SP constructs out of the schema of all data sources (Section 3). S can be used to verify the given query and to build the query. If a query cannot be constructed with the aid of global schema then it cannot be answered by CIS. However it is often possible that C can suitably modify the query by sacrificing some conditions or target information. SP can provide a query interface service to the customer to accept a query, e.g. to help the customer with the synonyms for the attributes (Section 3).

c. Data Read Privacy refers to the protection of data contents (values) belonging to a DS for accessing from other DSs, and C as well as SP. But for solving a
query sometimes data need to be shared with others. Data obfuscation has to be done before sending it to another party unless data sharing has been permitted by the data sources. Note there is a distinction between schema privacy and data privacy. Schema privacy will automatically imply data privacy, but the reverse is not true. If data is public for a DS implies that data access for that DS will be unrestricted. Yet precaution has to be taken to prevent data from being manipulated, i.e. written over. Either way maintenance of appropriate data privacy would encourage more data sources participate in CIS.

d. Data Write Privacy refers to how the organizations can protect their data sources from being written over by other organizations or the customer. Only valid transactions will be entertained. We are not distinguishing here between different write operations – insert, modify and delete. For simplicity we assume that write operations apply to data values only not to data definitions, i.e. schemas.

e. Result Privacy refers to the protection of the result which will be available to C from SP as well as the DSs. It is possible that C may not mind SP knowing the result, but does not want DSs to know their parts of the result; or it could be just the converse. In some cases C may not like to reveal the final query result as well as any intermediate computation to any third party. Specially note that when the activity is a write transaction then the result may consist of two parts, one part is for C and the other for one or more target organization.

f. Query Privacy refers to the protection of the customer query from SP and DSs. C is particularly interested to protect the sensitive parts of a query, e.g. while searching details of certain activities of a criminal within a given period of time, one may not like to disclose the identity of the criminal or the period of interest either to DSs or to SP. This would imply protecting query or query parts from respective DSs. Note we have already discussed in the previous section that syntax of a query has to be declared to SP, but the semantics (i.e. constants in a query) need not be disclosed.

 g. Query Distribution Privacy refers to the protection of knowledge of query distribution of SP from DSs and C. Note SP distributes the query to a set of DSs based on their availability and suitability. But protecting this information could be very crucial to the success of the query execution without violating any privacy. This protection works at two levels. At the first level there are DSs who are not involved in the query execution and hence may not have any idea about the ongoing query, particularly those whose data has no relevance for the query, either for the absence of relevant attributes or absence of relevant data, e.g. tuples in a relational database. The first one is taken care of by selecting the local schema appropriately. The second one may not always be so obvious, if the corresponding schema does not impose conditions explicitly visible at least to SP, e.g. a database can contain data for a specific region or country. At the second level DSs who are involved would not be informed about others’ participation. This information also may be kept from the client.

One could argue that these seven privacy issues encompass the whole range of protection required at the most basic level. The basic issues identity, data, result (read or write) and query are addressed here. The entire range of protections cannot be limited to just these seven issues. For higher granularity of protection any issue may need to be split generating more issues. For example, Data Write may be split into Date Insert, Data Modify and Data Delete. Similarly, the Schema, Data Read and Data Write issues could encroach upon the indexing issue (not considered in the present work) which basically allows better performance for query solution. On a similar ground whether do we treat identity issue at the level of IP address or that of name of the entities? Moreover power of these protections not only comes from the privacy issues per say, but also from the context or framework that is created. Take the example of multiple types of customers for protection. Whether additional protections can be superimposed by a given customer? 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.

4.2. Privacy Types

Let us look at the individual privacy issues at greater details. First consider the issue of identity privacy. It is the most important of all privacies. Two reasons can be cited. First it concerns each and every party. Second, it is the gateway for making accesses to a party. Here we have N+2 categories of parties – SP (service provider), C (client) and DS1, ..., DSN (Data sources). So (N+2)(N+1) one way communications (or more specifically access rights) are possible among these parties which result in a maximum of 2(N+2)(N+1) privacy types against identity privacy. Our objective is to find out permissible communications between any two parties. Strictly speaking if there are t categories of customers who have the same privacy requirements, then the total number of privacy options will be 2t(N+2)(N+1).

Local vs Global Privacy Template. The GDD consists of the global schema and global privacy template containing so called global privacy types. Correspondingly there are local schema and local
private templates for a specific query. The global privacy template concerns the privacy between the global partners: C_{dt}, DS’s and SP, whereas the local privacy concerns the local partners: c, ds’s and SP. C denotes a customer in general. A customer can belong to any of the t categories denoted by C_m, m=1,……,t. For a given query the customer C will share privacy restrictions imposed by different data sources applicable to the customers in category C_m to which C belongs. These restrictions are for access rights available to data sources and the service provider. However a specific customer is free to take decisions regarding access to its own query and result parts by other parties. The local and global schemas are defined in Section 3. Thus the local template is a subset of the global template. Note that this is not just a subset of global privacy types. The privacy types will themselves change as they are applied to different entities, e.g. a specific customer instead of the general customer, local schema in place of the global schema. Individually each local schema is a subset of the corresponding original schema. Since the treatment of global and local privacy types is identical we will consider only global templates.

**Enumeration of Privacy Types** – For determining the privacy types for different privacy issues there is a common phenomenon: symmetric vis-à-vis non-symmetric.

**Symmetric vs Non-symmetric Cases:** Privacy relationship, i.e. privacy constraints between any two players A and B will be only of two types, e.g. A protected from B = ‘Yes’ or ‘No’. Consider p players – B_1, …, B_p and A. The relationship between players \{A and B_{i}\}, …, and between \{A and B_{p}\} will be a total of 2^{p} types, p \geq 2, i.e. A protected from B_1 = ‘Yes’ or ‘No’, …, A protected from B_{p} = ‘Yes’ or ‘No’. Here B’s represent the players in the same category, e.g. data sources or schemas, meaning that their privacy concerns remain identical. The total number of privacy types between any B_i and B_j, i,j=1,…,p, is 2^{p(p-1)}. If we combine this with the relationship of A with B’s the number of privacy types rises to 2^{p(p-1)} + 2^{p}. These are the usual types which describe the most general situation, referred to as non-symmetric or asymmetric cases. But if 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, and so on, the total number of distinct types will reduce drastically to 2. Thus in the symmetric case with the relationship among the B’s reduces just to 2. For example, if p = 2 the number of types for the non-symmetric case will be 4, 4 and 8 respectively vis-à-vis the number of types for the corresponding symmetric cases are 2, 2 and 4 respectively. In the template definition for different privacy issues these will occur.

**5. Conclusions**

We have looked into the problem of collaborative computing for an information service. This gives a tremendous opportunity to organizations who want to do business together with their customers. Growth in concepts like distributed computing, cloud computing and crowd sourcing along with cryptography give rise to such ideas. The fundamental security issue has been discussed for a highly flexible (sometimes dynamic) security situation where different parties want different levels of privacy w.r.t. each other. The privacy issues and types have been used to model the 'privacy template'. The security problem is contemplated to be solved using a third party. The choice of the third party as an independent intermediary, a selected DS or a distributed mechanism would be an interesting problem. One could make role of SP more explicit as an intermediary by splitting the constraints into external and internal (Section 3.1). Based on our model one can go into making further inroads into the privacy model, such as indexing and work on privacy algebra [22]. Similarly further depth in privacy template could 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. For all these to achieve developing a Privacy Algebra might be necessary. More importantly one should try to come out with a feasible security protocol for a privacy template representing different privacy requirements given assumptions on cryptographic primitives and possibly certain restrictions on the kind of information activity requested. A formal model of evaluation and verification for the entire problem would be of interest to the security people.

**References**


