Reason based Access Control for Privacy Protection in Object Relational Database Systems

Emilin Shyni. C1, Dr. Swamynathan. S2, IACSIT Member

Abstract—A subject can execute a transaction only if the subject has selected or been assigned a role. While current information technology enables people to carry out their business virtually at any time in any place, it also provides the capability to store various types of information the users reveal during their activities. A key feature of our model is that it allows multiple purposes to be associated with each data element and also supports explicit prohibitions, thus allowing privacy offers to specify that some data should not be used for certain purposes. The RBAC model is based on the notion of users having an action status as well as an ascribed status. Privacy policy is to ensure that data can only be used for its intended purpose, and the access purpose should be compliant with the data’s intended purpose. We address this issue in the context of Object Relational Databases and propose different labeling schemes, each providing a different granularity.

Index Terms—Access Control, Data labeling, Reason, Purpose, Privacy, Status

I. INTRODUCTION

RBAC differs from Access Control Lists (ACLs) used in traditional discretionary access control systems in that it assigns permissions to specific operations with meaning in the organization, rather than to low level data objects. For example, an access control list could be used to grant or deny write access to a particular system file, but it would not dictate how that file could be changed. In an RBAC-based system, an operation might be to create a 'credit account' transaction in a financial application or to populate a 'blood sugar level test' record in a medical application. The assignment of permission to perform a particular operation is meaningful, because the operations are granular with meaning within the application. The model is applicable in situations where access policy information may be distributed and independently maintained, where access control requirements change in a highly dynamic way, where the actions of agents that request access to protected resources,

are important in rendering a decision on allowing the requested access or not, and where an access control checker can intelligently determine the authorizations that hold at any instance of time.

It is a natural expectation that the enterprise will use this information for various purposes, this leads to concern that the personal data may be misused. Many enterprises collect, store and use huge amount of personal information. The Federal Trade commission has shown that 97 percent of websites were collecting at least one type of identifying information such as name, e-mail address, or postal address of customers. The size, complexity, and dynamic nature of some distributed systems present particular challenges that demand that changes to access policies be made frequently.

The current database technology makes it possible to collect and store a massive amount of person-specific data, and the use of innovative knowledge extraction techniques combined [1] with data integration correlation techniques makes it possible to automatically extract a large body of information from the available databases and from a large variety of information repositories available on the web. Many companies such as IBM and the Royal Bank Financial Group use privacy as a brand differentiator.

The agent’s status level is used in rendering a decision on the agent’s access request. Another difficulty of privacy protection is that the comfort level of data usage varies from individual to individual. A key feature of the RBAC model is that a decision on an agent’s request to access resources is determined by considering the agent’s ascribed-status, the agent’s action status, and any additional conditions of relevance in answering the access request. In this article, we address this goal by presenting a comprehensive approach to purpose management, which is the fundamental building block on which purpose-based access control can be developed.

Our approach is based on intended purposes, which specify the intended usage of data, and access purposes are specified with respect to a hierarchical structure that organizes a set of purposes for a given enterprise. We address the problem of how to determine the purpose for which certain data are accessed by a given user. During the last few years, rapid technological developments especially in the field of information technology directed most attention and energy to the privacy protection of Internet users. On the other hand, these collected data sets are the most important tools for a wide range of studies. If an individual mentions that his/her data could not be used for a certain purpose, then his/her information is not accessible for the purpose.

II. RELATED WORK

This work is related to several topics in the area of privacy and security for data management, namely privacy policy
specification, privacy preserving data management systems and multilevel secure database systems. The W3C’s [1] platform for privacy preference (P3P) is an industry standard that intends to provide an automated method for users to gain access control over the use of their personal information collected by the web sites they visit. P3P provides a way for a web site to encode its data collection in a machine readable format known as a P3P policy, which can be compared against a user’s privacy preferences.

Even though P3P [2] provides a standard means for enterprises to make privacy promises to their users, P3P does not provide any mechanism to ensure that these promises are consistent with the internal data processing. Thus, P3P is merely a tool for making promises and does not help enterprises to keep their promises. The concept of Hippocratic database introduced by Agrawal et al amalgamates privacy protection in relational database system. A Hippocratic database includes privacy policies and authorizations that associate with each attribute and each user the usage purpose(s).

The enterprise privacy authorization language (EPAL) proposed by IBM is a formal language for writing enterprise privacy policies [3] to govern data handling practices in IT systems. An EPAL policy defines hierarchies of data categories, user-categories and purposes. User-categories are the entities (users/groups) that use collected data, and data-categories define different categories of collected data. Purposes model the services for which data is intended to be used. An EPAL policy also defines sets of actions, obligations and conditions. Actions model defines how the data is used, and obligations define actions that must be taken by the environment of EPAL. Lastly, conditions are Boolean expressions that evaluate the context. Privacy authorization rules are defined using these elements, and each rule allows or denies actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations.

Previous work on multilevel secure relational databases [4] also provides many valuable insights for designing a fine-grained secure data model. In a multilevel relational database system, every piece of information is classified into a security level, and every user is assigned a security clearance. The system ensures that each user gains access to only the data for which he has proper clearance, according to the well known basic instructions.

A major difference of this approach with respect to multilevel secure database is that in our approach each data element is associated with set of purposes, as opposed to a single security level. Also, the purposes form a hierarchy and can vary dynamically. These requirements are more complex than those concerning traditional multilevel secure applications.

Lefevre et al.[4] present an approach to enforcing privacy policy in database environments. Their work focuses on ensuring limited data disclosure based on that data providers have control over who is allowed to see their personal data and for what purpose. In their work they introduces two models of cell-level limited disclosure enforcement suggest an implementation based on query modification techniques.

Purpose based access control which has made a significant impact on many access control systems, greatly simplifies the specification and management of security policies within an enterprise. Permissions are assigned to functional roles within an enterprise and individual users to the necessary permissions by being assigned to a role or a set of roles. Most Purpose Based Access Control models include a role hierarchy, a partial order defining a relationship between roles and to facilitate the administration tasks.

Chen et.al [5] introduced the attributes associated with roles in order to enforce global constraints such as the principle of separation of duty. They discuss various attributes of roles, permissions, users and sessions and suggest practical way to specify and enforce constraints based on these attributes.

SBAC[3] introduced the agent’s action status is determined from a history of the deliberative actions performed by the agent. In SBAC, an agent is viewed as a rational entity that can, within certain constrains, choose the actions it performs.

### III. Preliminaries

A purpose describes the reason(s) for data collection and data access. Figure (1) gives an example of a purpose tree. A set of purposes, denoted as P, is organized in a tree structure, referred to as purpose tree and denoted as PT, where each node represents a purpose in P and each edge represents hierarchical relation between two purposes.

A model for controlling access to resources where permitted actions on resources are identified with roles rather than with individual subject identities. Intended purposes can be viewed as brief summaries of privacy policies for data, stating for which purposes data can be accessed. When an access to a data item is requested, the access purpose is checked against the intended purposes for the data item.

The intended purposes support both permissive and prohibitive privacy policies. An intended purpose consists of two components: allowed intended purpose (AIP) and prohibited intended purposes (PIP). This structure allows more compact and flexible policies in our model. We address this issue in the context of Object Relational Databases and propose four different labeling schemes, each providing a different granularity.

In Allowable Intended purpose the data providers explicitly allow accessing the data for a particular purpose. For example data providers may consider that his/her information can be used for marketing purpose without any further restrictions In prohibited Intended purpose the data providers strictly disallow accessing the data for a particular purpose. For example data providers may consider that his/her income information cannot be used for marketing purpose. In that case data provider’s income attribute is strictly prohibited to use for marketing purpose.
Let PT be a purpose tree and P be the set of purposes PT. Let x be a set of purposes in PT.

Ancestors(x), denoted by \( x^{\uparrow} \), is the set of all nodes that are ancestors of nodes in x, including nodes in x themselves.

Descendents(x), denoted by \( x^{\downarrow} \), is the set of all nodes that are descendants of nodes in x, including nodes in x themselves.

We use \( x^{\leftrightarrow} \) to denote the set of all nodes that are either ancestors or descendants of nodes in x, that is, \( x^{\leftrightarrow} = x^{\uparrow} \cup x^{\downarrow} \).

IV. ACCESS POLICIES

Evidently, how the system determines the purpose of an access request is as the decision of whether or not the access should be allowed is directly based upon the access purpose. There are various possible strategies to determine access purpose. First, the users can be required to state their purpose(s) along with the requests for data access. It requires complete trust the users and the overall privacy that the system is able to provide entirely relies on the user’s trustworthiness.

Lastly, the access purpose can be dynamically determined by the system, based on the current context. Role attributes are the pre-assigned, specific descriptions associated with each role.

The permissions to perform certain operations are assigned to specific roles. Members of staff (or other system users) are assigned particular roles, and through those role assignments acquire the permissions to perform particular system functions. Since users are not assigned permissions directly, but only acquire them through their role (or roles), management of individual user rights becomes a matter of simply assigning appropriate roles to the user; this simplifies common operations, such as adding a user, or changing a user's department. Then when the user activates the role, the values of the role attributes are loaded and made available to the access control system until the user deactivates the role. As such, the role attributes can be viewed as cached user information that is relevant to the specific roles.

The use of conditional roles provides great flexibility in that the authorizations are sensitive to both the user profiles and the system environments.

V. IMPLEMENTATION

A typical privacy policy for a data element includes purpose(s), retention and condition. It states that the particular data

The use of conditional roles provides great flexibility in that the authorizations are sensitive to both the user profiles and the system environments.
pip_code of the particular purpose pi is computed similarly by summing the encodings of every descendant and ancestor of pi with the encoding of Pi itself. Note that the last three columns of the pt_table can be automatically generated based on the first three columns using a simple procedure.

As described in Section 5, data elements are associated with intended purposes according to one of the intended purpose labeling schemes. When using the element-based labeling scheme: a table with n columns and x key column requires a purpose table with (x+2n) columns, x key column plus 2n columns for AIP and PIP (Example: IP_PURCHASE).

When using the row-based labeling scheme: a table with n columns and x key column requires a purpose table with (x+2) columns, x key column plus 2 columns for AIP and PIP (example: IP_ANALYSIS).

When the row-based labeling scheme requires additional tables for storing purpose metadata, the column-based labeling scheme for a table with n columns requires n entries in the privacy policy table (IP_COLUMN). Similarly, the Table-based labeling scheme for a table requires a single entry in the privacy policy table (IP_TABLE).

Intended purposes are encoded using the purpose encoding in the pt_table. As intended purposes have their implications, purposes are encoded using values from the AIP code or the PIP code instead of using their own encoding. Also, purposes can be combined by performing bitwise OR operations on the encoding of the purposes.

The policy document is a machine and human readable document, describes the different data items called person identification information, the general access purpose. While negotiating the privacy policy, the user views this document through a specially designed web application. The user can either accept the general privacy policy specified in the document or they can customize the policy to suit their privacy requirement.

### Access Purpose

Access purpose is intended to be the goal of the access purpose if and only if the access purpose is not prohibited by PIP and it is allowed. Access purpose is compliant with the intended purpose if and only if the access purpose is not prohibited by PIP. Thus, the purpose compliance check can be done with two bitwise AND operations as follows.

### SQL Query Modification Algorithm

Given the encodings of an access purpose, AIP and PIP, ap_code, aip_code and pip_code, respectively, the access purpose is compliant with the intended purpose if and only if (ap_code & pip_code) = 0, where & is bitwise AND operator and ^ is logical AND operator.

The query modification algorithm checks both the attributes referenced. As the attributes determine what data
items will be included in the result relation of a query, it may see enough to enforce privacy policy based only on the attributes in the list. The result of a query also depends on the predicates, and not enforcing the privacy constraints on the predicates may introduce inference channels.

EXAMPLE 1:

```
SELECT .full_Employee.Employee_pin PIN,
c.full_Employee.Employee.name NAME,
c.full_Employee.Employee.address.street STREET,
c.full_Employee.Employee.phone NO,
c.full_Employee.Employee.colour COLOUR,
c.full_Employee.customer.weight WEIGHT FROM Employee c
```

We have an object EMPLOYEE with four attributes id, name, street and city. Since the EMPLOYEE is configured at element level access level, the corresponding purpose metadata table IP_EMPLOYEE is joined with the actual table EMPLOYEE on its primary key EMPLOYEE_ID. The compliance check algorithm along with the query helps to filter out the data that complies with the stated access purpose.

The modified query looks like

```
SELECT .full_Employee.Employee_pin PIN,
c.full_Employee.Employee.name NAME,
c.full_Employee.Employee.address.street STREET,
c.full_Employee.Employee.phone NO,
c.full_Employee.Employee.colour COLOUR,
c.full_Employee.customer.weight WEIGHT FROM Employee c WHERE EMPLOYEE_PIN = IP_EMPLOYEE_PIN AND COMP_CHECK (2, PIN_AIP, PIN_PIP)
```

VI. Experimental Evaluation

In order to compare the overhead of the different intended purpose-labeling scheme precisely, selectivity of all data elements are set to 100 percent; i.e. all intended purpose label are set to allow for every access purpose. The performance of the system is improved by query cache. When the query executed for the first time, the query is modified and the modified query and its purpose are stored in cache memory (XML data). If the same query executed for the second time for the same purpose, the modified query is retrieved from the cache. This process improves the performance of the system. The following figure (6) shows the response times of queries against relations with various labeling schemes. As shown, the response time increases the granularity of labeling schemes finer. For the element-based labeling scheme, the number of attributes accessed by queries is also a major factor. As the element-based labeling scheme requires a compliance check for every element a query is accessing, the overhead of compliance checks becomes significant as the number of accessed attribute increases. This shows that the use of both AIP and PIP does not introduce much overhead, compared to the case where only AIP is used.

The following figure (7) shows the result of our experiments with two different sizes of purpose trees. The first purpose tree has five nodes with the height of two, and it requires five bits to encode all possible intended purposes. The second purpose tree has 14 nodes with the height of five, requiring 14 bits all possible intended purpose encodings. As the result shows, the size of purpose tree does not make any substantial difference in either the object or element-based labeling scheme.

The figure 8 shows the storage overheads of the element-based labeling scheme with relations of three different object sizes. These relations have the same cardinality of 100k.

VII. Conclusion and Future Work

As part of this work, the access control model for privacy protection based on the notion of purpose. A novel purpose based access control method in ORDBMS was analyzed, implemented and tested. It is the efficient method for determining access purposes, which uses the notions of role attributes and conditional roles. For query modification various implementation issues are suggested. The conventional access control model such as role based access control, deals more on security not the privacy protection. This model supports four levels of data labeling; relation or table based labeling, attribute or column based labeling, tuple or row based labeling and element or individual cell based
labeling.
To improve our current implementation, we plan to investigate techniques to implement the proposed work in the DBMS internals. This will reduce the redundant query parsing by both Purpose based access control and DBMS. Our future work includes Dynamic Purpose-based Access Control (DPBAC) and in the other access control system.

REFERENCES

Mrs. C. Emilin Shyni is young and dynamic Computer Professional with over 12 years of teaching experience. She is a research scholar in Department of Computer science and Engineering, CGE campus, Anna University Chennai. She has around 12 publications in conferences.

Dr. Swamynathan is currently working as an Assistant Professor in Department of Computer Science and Engineering, CEG Campus, Anna University Chennai. His area of interest include Distributed computing, Artificial intelligence and Advanced databases. He is currently guiding ten Ph.D. scholars in different topics. He has around 30 publications in international conferences and journals.