A Implementation of Object Oriented Database Security

Conference Paper · September 2007

DOI: 10.1109/SERA.2007.17 · Source: IEEE Xplore

A IMPLEMENTATION OF OBJECT ORIENTED DATABASE SECURITY

Premchand B. Ambhore,B.B.Meshram,V.B.Waghmare
Lecturer in Computer Engineering
Dr.P.D.Polytechnic, Amravati, M.S (India)
pbambhore@yahoo.com, pbambhore@rediffmail.com

Abstract: - This paper is to address high-level authorization specifications and its efficient implementation in object oriented database scenario. Premchand Ambhore is with higher and technical education, maharashtra state, india, E-mail:pbambhore@yahoo.com . The rest of the paper is organized as follows. Section 2 specifies object oriented database system; section 3 incorporates authorization rules into database and section 4 maps the specification into a logic programming language and implements the language. Section 5 concludes the paper. The integration of object-oriented programming concepts with databases is one of the most significant advances in the evolution of database system, therefore the problems involved in providing data security for object oriented databases have been receiving very great attention from researchers. As a matter of fact, data are the most sensitive part of any system and its loss or compromise can have disastrous consequences. We discuss three different types of access: Discretionary access control (DAC), Mandatory access control (MAC) and role based access control (RBAC) in Relational DBMS and object oriented data bases. We have identified some of the issues and current proposals; current security models in object oriented database systems and outline possible direction issues for future research.

Keywords – Semantics, structural, and dynamic, data model, access control, database security, authorization policy, and logic-based specification.

I.INTRODUCTION

In the history of data processing to date, we can clearly distinguish two particular trends: the trends towards data independence and data usability in every area involving user / system interaction, we can trace a steady improvement in the quality and level of the human interface. The trend is always away from the irrelevant complexities of the underlying machine and physical data organization and towards a higher and more causal user oriented level of expression. At the same time, the computation power kept on increasing in a wide range as a result of computer networks and e-commerce systems. These evolutions have had a huge impact on the design of security systems in the DBMS. Initially, the security system was not an independent module of the DBMS. It relied heavily on the security system of the underlying operating system. Then the need for more powerful controls, for a finer grain of protection, and the inadequacy of identification / authentication schemes led to the design of security systems as a component of a DBMS.Data security is intrinsically two fold; it includes the privacy protection and soundness of the stored information. Privacy control dictates the access control so as to restrict users to retrieve only the data structures or values for which they receive authorization. Soundness of the stored information dictates integrity, which basically encompasses two major features; the updates synchronization of shard data by concurrent users (External integrity) and correctness of data modification depending on predefined constraints (Internal integrity). Thus data security mechanism must ensure that a user is authorized to have access to data for the performance of specified operations, that any changes resulting from a user's alternation of the database are valid without affecting the consistency of the whole database (13,15). In the past several years, several new types of database have moved out of the academic world and have been released as commercial products. These new types of database are commonly referred to as next generation database and include object-oriented, object-relational active and deductive databases. Each of these types of database offers new features when compared to the more traditional relational databases. In turn, these new databases require new security models to correctly regulate access to their data.

Security Challenges In Next-Generation Database: Both object-oriented [1] and object relational database need the ability to protect the object features of the given database. The issue of how to protect a class definition can be seen as similar to the problem of protecting the definition of a table with a relational database. Class definitions allow the possibility by altering or deleting a class definition. Implication of data is also possible if a malicious user cam view the names and types of a class attributes or the interface definitions for a class methods. Controlling the ability to create, modify, or destroy a class definition, while limiting the ability to read the definition of a class can control inference can protect against corruption. Issues such as inheritance and versioning complicate this matter in databases. For example while inheritance offers many advantages, it also creates several problems in designing a security model for general-purpose object oriented database system. Given a user has access to a parent class, is there any implicit permission or restriction on access to a child class As a result, careful consideration must be given to defining the semantics of inheritance when security is a concern [1].

Similar problems arise when attempting to protect various versions of a class. Complex objects pose a problem similar to the one posed by inheritance and versioning. For any given complex object, should any implicit authorizations for a sub-object be given based upon an explicit authorization for the parent class? Methods also provide several challenges. The obvious need is to be able to control execution of the method. A subtler problem arises when one method calls others. Again, is there any implicit authorization for the caller function has been given? More challenges arise if the OODBMS supports the concepts of information hiding. All these issues are discussed with reference to DAC, MAC or the combination of both or RBAC for protection. A DAC specifies the access types of users on data, and the rules whereby users can, at their discretion, grant and revoke their authorizations of other users.A MAC policy specifies the rules whereby users can obtain direct or indirect access to classified data, and the rules for sanitizing and reclassifying data. These policies apply only to multilevel databases, which are databases that contain information of different classification. A RBAC provides mechanisms for the enforcement of the least privilege and separation of duties.The remaining part of the paper is organized as below Section II describes Discretionary Access Control (DAC) issues and approaches in RDBMS and explains structural, behavioral and semantic based access control and models in OODBBMS. Section III deals with Mandatory Access Control (MAC) issues and approaches and MAC security models, Section IV explains Role Based Access Control (RBAC) issues and approaches & RBAC security models. Section V concludes the results and future research directions.

II.DISCRETIONARY ACCESS CONTROL

DAC allows privileges to be granted to other users by the object owners (7). In DAC, the way in which individual users manipulate specific objects is specified through explicit access rules. They are fundamental to operating systems as well as database system.

A.DAC in RDBMS DAC is based on granting and revoking privileges. Informally, there are two levels for assigning privileges to use the database system:1. The account level 2. Relation level. The granting and revoking of privileges generally follows an authorization model for discrenationay privileges known as the access matrix model, where the rows of the matrix M represent subjects (users, accounts, programs) and the columns represents objects (relations, records, columns, views, operation). Each position M (i,j) in the matrix represents the types of privileges (read, write, update) that subject I holds on object. The owner account holder can pass privileges on the relation. The owner account holder can pass privileges of owned relations to other users by granting privileges to their accounts.

In SQL the following types of privileges can be granted on each individual relations R : select privileges on R, modify privileges on, r, reference privileges on R we can also specify privileges using view (13).We can propagate privileges using the grant option such as GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 WITH GRANT OPTION. We can also REVOKE privileges by REVOKE command such as REVOKE SELECT ON EMPLOYEE FROM A3

B.DAC- Issues & Approaches in OODBMSA well known access control principle is to organize subjects into control groups, based on their roles in an organization. This makes it easier to grant and revoke authorizations to entire groups of subjects at a time. The role lattice is used to define such groups, and implicit authorizations propagate from the top to the bottom of the lattice. Let us consider the following important issues related to DAC in-O environment [15,33]. Granularity of Authorization. In the authorization system for an OODBMS, the smallest unit of authorization may be a class, an object instance or an attribute. Operations (Access Types): We assume the following set of operations in the authorization system on the protected object: read, write delete, execute and create. The operations are partially ordered such that access right to operations of higher order implies access right to operations of lower order. The assumed order is: read<execute<write<delete<create. Authorization Types: There are two types of authorization in an OODBMS: full and partial authorization. A user with full authorization right for a seat of operations on an object has the same operation rights to the components of the object. In the case of partial authorization, access to an object does not extend to the components of the object. Similarly access right to a set of objects does not imply right to access members of the set. Visibility From Above: An object may have many descendants or equivalently, an object may have many ancestors. In this case, if a user is authorized to access an object, (s) he must also be authorized to access all information of all descendants of that object. On the other hand, lathe user must not be able to access the information of the ancestors of that object. In other words, the data must only be visible from above and not be visible from below. Visibility From Below: In class / sub-class hierarchy, the access to a subclass implies access to those attributes values of the super classes that correspond to the subclass because of inheritance property. This property is called visibility from below. Thus when users have full authorization rights to some operations on an object-instance of a subclass, they have the implicitly the same authorization rights to relevant object-instances

of the super classes. Grouping: The DAC models not only determine which subjects may obtain access to which objects, but also determine grant and revoke authorization policy for users and groups.

C.DAC in Structurally OODBMS

In structural approaches, an access control request poses the basic question. Is subject A allowed to read / write / delete object O? Let us see the details on how this question is answered (30,20).Subject To Object Access Control: The basic idea is to group subjects into access control groups and grant authorizations in terms of access types such as read, write, and delete. These access types are usually ordered such that the authorization for the right may include others. This is accomplished by utilizing implicit authorizations along an access / authorization type lattice.Inter-Object And Interaobject Access Control: As per our observations, some of the issues are difficult to categorize clearly as inter-object or intra-object, or both. For example, the inheritance of attributes is an inter object issue, since it involves at at least two objects, but at the same time is also an intra-object issue for the object that is inheriting the attributes. Variable Granularity For Access Units: In structurally object oriented database systems, the access control mechanisms would have to flexible enough to support varying granary of access units. For example, it may be desirable in some applications to have fixed graned access control at the level of the individual attributes of an object. But it may also be desirable to grant access / authorization to large substructures (such an entire objects or composite objects) as a single unit. Class Hierarchy And Structure Inheritance: While inheritances offer many advantages, it also creates several problems in designing a security models for OODBMS. As a result, careful consideration must be given to defining the semantics of inheritance when security is concern. In particular, the following questions need to be addressed. What effect does allowing implicit authorizations in the class hierarchy have on the reusability of classes? On query processing? What should be the semantics for the inheritance of structure (attributes) among classes when subject have different authorizations on these classes? Our First Option. The implicit authorizations should be given to the creator of a class on all instances of a subclass derived from the class. Queries rooted on the class and spanning lower subclasses can be evaluated successfully as the required authorization can be obtained. However this approach makes the classes to independent making their potential for reuse very low. Our Second Option. The second option would be to prohibit implicit authorizations from classes to instances the reusability of classes, but this benefit comes at the cost of query failures.

D. DAC Models In Structurally OODBMS

We will explain two DAC medals [19,20] in structurally OODBMS namely (1) DAMKLES access control models [2] LPGSF access control model.

A.Damolkes Access Control Model Dietrich, Hartig, and Pfefferle developed a DAC model for the DAMKLES system. The model supports two types of object. Design object data model (DODM) objects and DODM r relationship. In DAMOKLES data model (DOM) every object is further broken down into smaller access units called protection objects (p-objects). These projects are: the descriptive part D (objects properties), the structural part S, (composite object version part V, (the objects versions) and the role part p consisting of a relationship's roles that is participating objects. The access control allows a users to grant a privilege to other users if user is the owner of the object. The ownership is transferable but is not transitive. In the model, least access must be explicitly authorized. The model applies exist, read, write, delete, access privileges. The access privileges are partially ordered as: exist<red<writ<delete. The triple (S,O,R) specifies a single assigned an access privilege r to an object o. There are two types of authorization. Simple and complex. In the simple authorization a p-object o is D, S, V or p part of a complex object. In the complex authorization, a privilege is given to an entire object including, its components which belong to the same owner. Thus if an object actually contains components of various owners, a subject has to get permission from all of them to be able to work with entire object.

B. LPGSF Access Control Model Larrondo-Petrie, Guides, Song, and Femandez developed an access control model (LPGSF) model. The model is based on a set of policies that define authorization inheritance through class hierarchies. An access rule is a tuple (S,O,R[C] which state that C is a condition that must be satisfied for the object O that the subject S can use the privilege R. The model enforces the following policies: Inheritance policy, visibility from below policy, visibility from above policy and overriding policy. For overriding policy, we have the following order (S,O-R)>(S,O,+R)>(S,O-R)>(S,O,+R)

E.DACSEMANTICBASED ACCESS CONTROL

The following factors are considered in DAC Semantic Based Access Control. Subject to Object Access Control: Access control would involve authorization to invoke an initial method in a chain / tree of method invocations. We describe three approaches for subject to object access control that have been reported in the literature. In the first approach, associated with every object is an access control list (ACL) and an object owner. The owner of an object controls through the ACL the other principals that may invoke the operations (methods) defined for the object. In the second approach, access group based on user roles are defined with the help of a user role definition hierarchy (URDH). A node in the URDH represents an access group. Based on the access control requirements, the public ally accessible methods are assigned to nodes in the URDH. A subject/user belonging to a particular node in URDH will be allowed to invoke only those methods assigned to that node. A third approach users the notion of interface objects. Every database object is associated with a collection of interface objects. An interface object supports only a subset of the total methods in a database object. Subjects are allowed to interact with the database object. Subjects are allowed to interact with the database objects only by invoking methods defined in their corresponding interface objects. Inter-Object and Intra-Object Access : If we wish to control the visibility of a method M, then we should restrict the number of client methods that can invoke M. These approaches associate a set < invokers > with the definition of every method M. This set contains the names of methods and classes to which M is made visible. The need for access control resurfaces even within the boundary of an object. It must be recognized that access control and integrity mechanisms are closely liked. This is because methods modify the states of objects and we often enforce access control on method invocations. Integrity after all, is concerned with the improper modifications of data. Method To Attribute Visibility: For some applications, it may be desirable to allow only certain methods in the object (or class) to access a local attribute. For example, in a military application an attribute 'Target-Coordinate' may be updated ensures that the new coordinates are not erroneous. An obvious way to achieve this would be for every attribute to maintain a list of methods that are allowed access to the attribute. Method-To-Method Visibility. Within an object boundary, we may want to restrict the visibility of local methods to each other. A gain an obvious way to accomplish this would be for every method to maintain some list. More complicated data structures are worth investigating especially if the notion of implicit authorization can be applied.

F. DAC Model In Behaviorally OODBMS

In an access control model (26) for behaviorally OODBMS, we are concerned with how the subject gets permission for methods in the sequence of calls. In this survey study, we consider the following DAC models in behaviorally OOBDMS namely.

1. Iris Access control model.
2. Bettino data hiding model.
3. Iris Access Control Model.

In the Iris data model, the access control model is based on a concept of the function call control and evaluation of the call. The action entities are stored functions, derived functions, generic functions, guard functions and proxy functions. The access control is implemented using functions that are allowed to be called by a user. An access rule is a triple (U,F,T) where u is the user name, f is a function or a set of functions, and t is the type of the argument to the function. Note that the access of u is limited by the set of functions f the user is allowed to evaluate. Functions f can be either stored or derived. Access to derived functions can be static or dynamic. If a user u has a dynamic access to the derived function, f, u must have a permission to call all the underlying function. For a static access, the user u does not need to have call privileges to the underlying fuctions. Note that the creaor of the derived function must have call privileges to all the underlying fuctions. The Iris access control modlel supports time-dependent and content-depended authorizations using guard and proxy functions, respectively. This model also supports the concept of ownership and DBA. A user who creates a function is its owner DBA or a group of DBA has The Privileges implied by the owners of function.

2. Bertino Data Hiding Model

Bertinomodel uses methods as a tool to control the access to objects (27). The access control can have two levels. External level an internal level. For the external access control, a triple (U, O, M) indicates that a user U can call the method M in the object O. The internal access control is content dependent and is implemented as part of methods. Users can have privileges to public methods only. For instance, if a user u invoke a method M1 on O which in turn call M2 then M1 has to be public and user has to have execution right on that. If M2 is public, the entry (U, O,M2) must exist. If M2 is private, M1 must belong to the invocation scope of M2 the model introduces the notion of protection mode.

III.MANDATORY ACCESS CONTROL

A.MAC In RDBMS: DAC is the main security mechanism for RDBMS, but in many applications, an additional security policy is needed that classifies data and users based on security classes. This approach known as MAC for Multilevel security would typically would be combined with the DAC mechanisms. The need of multilevel security exists in government, military and intelligence applications, as well as in many industrial and corporate applications. Typical security classes are top secret (TS), secrete (S), confidential ©, and unclassified (U) where TS is the highest level and U the lowest level. For simplicity TS >S>C>U The commonly used for multilevel security known as the Bellk-LaPadula model classifies each subject (user, account, program) and object (relation, tuple, column, view, operations) into one of the classifications TS, S, C, U. B. MAC In OODBMS

MAC is usually implement in the form of multilevel security [19,22,29,31,14].

MAC Security Policy The MAC security policy is formulated as given belowDatabase Operations Dependent MAC Security: Database operations are implemented by messages that activate corresponding methods that implement them. Our message paradigm is of the form: (Sender , Receiver, Method-selector (arg1…….argn) From the point of view of information security, our concern is the passing of security relevant information within the message to evaluate whether or not a response to some message would or would not violet security. We group database operations into two categories: those with side effects and those without side effect. Operations with side effects and those without side effect. Operations with side effects either alter the state of objects in the database or are responsible for the generation of new objects in the database, while those without side effects merely return the state of database objects that they operate on. Context-dependent MAC: Context-dependent access control refers to a combination of items with restrictions places on objects or attributes that can be retrieved together. In our case, we allow three types of context.Context-dependent in which we hold a list of names plus the values for which the given classification are requires.Name dependent classifications where for the specified name we give a minimum classification level. Content dependent where retrieval of two objects or attributes must together be classified at some specified security level.

B. MAC Security Models

Some of the proposed security models for OODBMS are discussed briefly:

1. Sorion Security ModelSorion is a security model proposed by Thurainsingham to incorporate a secure access control into the ORION model The Sorion model is defined in terms of subject, entities, and access privileges. The model allows the read, write and execute access privileges. The access to an entity is controlled by checking security level of the entity, which is computed according to the security rules of the system.

2. Jajodia-Dogan Security Model Jajodia-Dogan (6,12) proposed a security model for OOBBMS that control access by using the encapsulation characteristic of o-o system. The model control information flow by filtering the message transmitted between objects. That is why it is also called the message filter model. The message filter intercepts every message exchanged between objects. The message filter decides how to handle the message based on the security levels of the sender and receiver, and information encountered in a chain of method executions. The list of possible actions that the message filter can undertake includes: leaving the message unaltered, blocking the message and enforcing restriction on the execution of the method invoked by the message.

IV. ROLE BASED ACCESS CONTROL

Each organization has unique security requirements, many of which are difficult to meet using traditional access control such as MAC and DAC. Access control decisions are often determined by the roles individual users take on as a part of their job. The roles are usually specified by user responsibilities, and qualification. The usual grouping mechanism of DAC can be used to implement roles A. RBAC in RDBMS For a user to do anything, DBA
need to grant the user one or more system privileges. A system privilege is a privilege defined by the RDBMS that allows a user to perform a certain task, such as creating a table. Privileges are frequently bundled together into roles. System privilege: To grant a system privilege or a role to a user, use the GRANT Command as follows: GRANT {system_priv|role} [,{system_priv|role}] TO {user | role| PUBLIC} [, {user | role |
PUBLIC}] [WITH ADMIN OPTION]

Commonly granted privilege are create session, connect, resource, create table, create view, create sequence, create procedure, create trigger, create synonyn. You can list as many as privileges and roles in the command as you need.

B. Issues And Approaches IN OODBMS Roles And Information Portion: For a user authorized for a given role(s), only the information accessible via the role(s) is available to user. Viewed this way, and for a protected database, a role is a window into the database. The information visible / accessible via such a window is a context by itself. In general, roles partition database information into context, with each context and disjoint information context respectively. The intersection of information contexts is a matter of security policy [3,18, and 23]. Roles and Separation of Duty: Separation of duty is applied where several people are required to perform a given task such a task would be broken into subparts which are then assigned to different people. Every individual is then required to perform only one of the subtasks with the restriction that none of the individual can perform more than one subtask. The 00 paradigm is suitable for this scheme of role based protection in which we have roles authorized to execute specific methods associated with some objects. The resulting effect is the distribution of an object's interface across roles. Separation of duty requires audit information to ensure that before subjects are allowed execution; they have not participated in the processing before. However searching for such information in the audit record can be very costly. Hence we use object history to keep track of it own audit information [24,25.]

C. RBAC Security ModelsSome of the proposed RBAC security models are discussed briefly [10,11]

1.The Cleark And Wilson Model The Clark and Wilson model for commercial security proposes to model the secdurity requirements of commercial environments, which stress integrity more than secrecy. Thus apart from separation of duty, there are requirements such as all database items being constrained data items (CDIs) that only certified transformation procedures (TPs) manipulate the CDIs, that the TPs are certified to take CDIs from one correct state to another, that there are verification procedures (IVPs) that assure the integrity of code which manipulates the CDIs, etc. In general, all data transformations are required to be designed as well-formed transactions (WFTs) where a WET is a program that has been certified to maintain the integrity of the data it manipulates. Separation of duties requires the association of TPs with users and associated CDIS, which leads to relationships of the for (VID, (Tpi (CDI1 CDI2…))…) where VID is the user identity.

2. Nyanchama And Osborn Model Matunda Nyanchama & Sylvia Osborn combine concepts of role based protection and object-oriented databases to specify and enforce separation of duty as required for commercial database integrity. Role essentially partion database information into access context. Methods of the class associated with the database object also partion the object inter face to provide windowed access to object information. By specifying that all database information is held in the database objects and authorizing methods to roles, they achieve object inter face distribution across roles for processing in the commercial system and they can design objects and distribute their associated methods to different roles. In this model, method execution is a part of a transactional process that reads history information, uses it along with authorization information and updates the history.

V. CONCLUSIONS

Object oriented database have moved out of the academic world and have become available as commercial products. The new features which these types of database require new and different approaches to securing the data held within them and each next generation database provides its own flavor of security as the relational database models. Discretionary access control issues such as granularity of authorizations, access types, authorization types, visibility form above & below, grouping are identified. We have identified some of the areas that warrant further research including authorizations based on separation of duties, multiple approvals & object contents.

We have addressed the issued of mandatory security within an object oriented database with respect to database operations. We emphasized the need for retention of the mandatory security specifications of object involved in an operation to abe able to determine the security specifications of the objects resulting from the operations. We have also identified the problems caused by inheritance for development of a security model for such systems when access controls are defined to exploit the inheritance mechanism.We have also discussed the enforcement of information partion and separation of duty using both role based protection and OO database principle. Using OO technology we can incorporate audit information in the object structure and impose conditions on method execution that must access the history execution and update it on completion of execution.

6. REFERENCES

[1] B.B. Meshram & T.R. Sontakke “ SYSTEMATIC Management of Object Oriented Project”CSI Communication, May-June 2001.

[2] B.B. Meshram & T. R. Sontakke “Object Oriented Database schema Design”, 7th International Coference on Object Oriented Information System, Calgary, Canada 27-29 August 2001

[3] P.B.Ambhore, B.B. Meshram & V.B.Waghmare “tool for development of javaapplication using Object Oriented modeling and desing”, “National Coference on “Trends in computational techniques in engineering”October 15-16,2004

[4] E, Bertino, F. Buccafurri, E. Ferrari and P. Rullo, “ A Logic-based Approach for Enforcing Access Control “. Computer Security,vol.8, No.2- 2, pp 109-140,2000.

[5]E, Bertino, B. Catania, E. Ferrari and P. Perlasca, “A Logical Framework for Reasoning about Access Control Models.” ACM Transactions on Information and System Security, Vol. 6, No.1pp71-127,2003.

[6]C. Bettini, S. Jajodia, X. S. Wang and D. Wijesekera, “Provisions and Obligations in Policy Management and Security Applications.” Proceedings of the Very Large Databse Conference, pp502-513,2002.

[7]S. Jajodia, P. Samarati, M.L. Sapino and V.S. Subrahmanian, “Flexible Support for Multiple Access Control Policies.” ACM Transactions on Database System, Vol. 29, No.2, pp214-260,2001

[8]N. Li. B. Grosof and J. Feigenbaum, “ Delegation Logic: A Logic-based Approac to Distributed Authorization”. ACM Transactions on Information and System Security,Vol. 6, No.1, pp128-171,2003.

[9]L. Wang, D. Wijesekera and S. Jajodia, “ A logic-based framework for attribute based access control”, Proceedings of the ACM Workshop on Formal Methods in Security Engineering, pp45-55,2004.

[10]T.Y.C. Woo and S.S. Lam, “ Authorization in Distributed systems: A Formal Approach”. Proceedings of IEEE Symposium on Research in Security and Privacy, pp33-50,1992.

[11] M. D. Abrams.Role-Based Access Control Position paper. In proceedings of the 17th National Computer security conference, volume 2, Page 491, Baltimore, Maryland, October 1994.

[12] E. Bertino, S Jajodia, P. Samarati, Database Security: Research and practice, Information System, 20(7) (1996) 537-556.

[13] E. Bertino, C. Bettini, E. Ferrari, P. Samarati, Supporting periodic authorization and temporal reasoning in data base access control, in proc. 22nd Intl. Conf. On very Large Data bases (VLDB;96), Bombay (India), September3-6, 1996(Morgan Kaufman Publishers).

[14] E. Bertino, P. Samrati, S. Jajodia, An extended authorization mode, IEEE Trans. On knowledge and data Engineering 9(1) (January/February 1997) 85-101.

[15] K. R. Dittric, M. Hartig, and H. Pfefferle. Discretionary Access control in Structurally Object Oriented Database Systems. In C.E. Landwehr, editor, Database Security II: Status and prospects. IFIP, North-Holland, 1989.

[16] G. Matunda Nyanchama,Object Oriented Database Security. Masters thesis, Dept. of Computer science, The University of Westerm Ontario, London Canada, August 1991.

[17] M. B. Thuraisingham, Mandatory Security in object Oriented Database System. In OOPSLA’89 Conference Proceedings. ACM SIGPLAN Notices, 1989.

[18] D, Ferralio, J. Cugini, and D. Kuhn. Role-Based Access control (RBAC): Features and motivaytiond. In Proceedings of the 11th Annual Computer Security Applications Conference, Pages 241-248, New Orleans, USA, December 1995. IEE Computer Society Press.

[19] G. E. Gajnak. Some Result from the entity / Relationship Multilevel Secure DBMS Project. In T. F. Lunt, editor, Discussions of topics presented at a workshop held at the

vallombrosa, Conference and Retreat center, Menlo Park, CA May 1988, Research Directions in Database security, Pages 173-190. Springer – Verlag, 1992.

[20] Pierangela Samarati, Elisa Bertino, Alessandro Ciampichetti, and sushil jajodia, Information Flow Control in Object-Oriented System. In JEEE Transactions on knowledge and Data Engineering, VOL 9, No 4, July/August 1997, PP 524 Castano,s., Fugini, M. Martella, G., and Samrati, P. Database Security, ACM Press, 1995.

[21] Oracle SQL Reference, Database Object Priviledges.

[22] M. S. Oliver and S.H. Von Solms. Taxonomy for secure Object Oriented Databases. ACM Transactions on Database Systems, 19 (1):3 46, March 1993.,

[23] N. Boulahia cuppens. D. cuppens. Gabillon and K. Yazadanian. Decomposition of Multilevel Objects in an objects-oriented Database. In Computer Security ESORICS 94, Third European Symposium on Research in computer security, Volume 875 of Lecture Notes in Computer science pages 375-402. Springer Verlag, November 1994.

[24] F. Cuppens. A Model Logic Framework to solve Aggregation problems. In C.E. Landwehr and S.Jajodia, editors, Database Security V, pages 315-333. Elsevier Science Publishers B. V. (North-Holland) IFIP,1992.

[25] M. D. Abrams. Role based Access control position paper. In proceedings of the 17th National Computer, Volume 2, Page 491, Baltimore, Maryland, October 1994.

[26] R. Ahad. J. Davis, S. Gower, P. Lyngbeak, A. marynowski, and Eonuegbe. Supporting Access Control in an Object-Oriented Database Language. In Procedings of the 3rd International Conference on Extending Database Technolog, EDBT 92 volume 580 of Lecture Notes in computer Science, Pages 184-200, Vienna, March 1992. springer-Verlag.

[27] P.E. Amman and R. S. Sandhu. Implementing Transaction control Expressions by Checking for Absence of Access Rights. In 8th Annual computer Security applications Conference, Pages 131-140. IEEE computer society Press, 1992.

[28] E.B. Fernadez,, J. Wu. And M.H. Femandez. User Group Structures in Object-Oriented Database Authorizations. In J. Biskup, M. Morgenstem, and C. E. Landwehr, editors, Database Security VIII (A-60), pages 57-76. Elsevier Science Publishers B. V. (North-Holland) IFIP, 1994.

[29] S. Jajodia and B. Koga. Integrating an Object-oriented Data-Model with Multilevel security. In Proc. 1990 IEEE Computer society Symposium on Research in security and Privacy, Pages 76-85. IEEE Computer society Press, May 1990.

[30] L. G. Lawrence. The Role of Roles. Computers & Security, 12(1):15-21, Feb. 1993.

[23] M. Nyanchsms and S. L. Osbom. Role based Security:Pros, cons & Some research Directions. ACM SIGSAAC Review,2(20:11-17, June 1993.

[31] T. C. Ting, S. A. Dermurjan, And M. Y. Hu. Requirements Capabilities and Functionalities of User-Role Based Security for an object-Oriented Design Model In C. E. Landwehr and S. Jajodia, editors, Database Security V:status & Prospects pages 275-296 North-Holland,1992.

[32] E. Bertino, S. Jajodia, and P. Samarati. Access control in object-oriented Database Systems: Some Approaches and Issues. In N. Adam and B. Bhargava, editors, Advanced database. In J. Biskup, M. Morgenstem, and C.E. Landwehr, editors, Databse Security VIII(A-60), pages 199-222 Elsevier Science Publishers B.V. (North-Holland) IFIP<1994.

[33] Elisa Bertino. Data Hinding and Security in Object-Oriented Databases systems. In P. P. Chen and R. P. Van De Riet, Editors, Data & Knowledge Engineering, pages 1-29. Elsevier Science Publishers B. V. (North-Holland) IFIP, 1994.

[34] Elisa Bertino. Data Hiding and Security in object-Oriented Database In F. Golshani, editor, proceeding of the eight International Conference on Data Engineering, Pages 338-347. IEEE Computer Society Press, 1992.

[35] Elisa Bertino and Sushil Jajodia. Modeling Multilevel Entities Using Single Level Object. In Proceedings of the Deductive and Object-Oriented Database. Third International Conference, DOOD'93, Volume 760 of lecture Notes in computer Science, Pages 415-428, Phonix, Arizona, USA, December 1993. Springer-Verlag.

[36] Elisa Bertino and Pierangela Samarati. Research Issues in Dictionary Authorizations for Object bases. In OOPSLA'936 Workshops on Security for Object-Oriented System, Pages 183-199. ACM SIGPLAN Notices, October 1993.

[30] L. J. Binns. Inference and cover stories. In B. M. Thuraiisingham and Landwehr, editors,Database, Security VI, pages 169-178. Elsevier Science Publishers B. V. (North-Holland) IFIP, 1993.

[37] S. Abiteboul, R. Hull, and V. Vianu, "Foundation of Databases," pp.563-571, Addision-Wesley Publishing company,1995.

[38] E.Bertino and H. Weigand, "An approach to authorization modeling in object-Oriented database system,"Data and knowledge Engineering vol. 12,no.1,pp.1-29,1994.

[39] H. H. Brijggemann, "object-oriented authorization" advances in Database System-Implementations and Applications, CISM 347, pp. 139-160, Springer-Verlag, 1994

.