PRIVACY PROTECTION ALGORITHM FOR ENVIRONMENT BASED DYNAMIC ACCESS CONTROL MODEL TO ACHIEVE DATABASE SECURITY

Elusoji Adekeyede Akorede & Haastrup Adeleye Victor PhD
Computer Technology Department
Yaba College Of Technology
Yaba, Lagos-State, Nigeria.
elusoji872@yahoo.com, victleye@gmail.com

Onukwugha Chinwe Gilean
Computer Science Department
Imo State Polytechnic
Imo -State, Nigeria.
onukwugha2000@yahoo.com

Odiketa Juliet Chioma
Computer Science Department
Federal Polytechnic
Idah, kogi- State, Nigeria.
chiomie@yahoo.com

Akanji A.Wasiu.
Computer Technology Department
Lagos State Polytechnic
Lagos-State, Nigeria.
akanjiwasiu2005@yahoo.com

ABSTRACT

With the advancement in distributed computing and collaborative software technologies, information sharing and privacy related issues are gaining interest of researchers related to digital information creation, management, and distribution. With the fast growing nature of enterprise business especially with the emergence of information technology, we are moving towards the era where database systems have become mandatory for the organizations to implement. Because of this, it has become very important to specify such access control model for the database systems in organizations that must ensure the security of information but at the same time dynamic. Conventionally, access control models stress on pre-defined users for which access level is pre-determined by the database administrator. Considering the need of today's business, that has become borderless, and most of unknown users also attempt to access the information. This paper, show the basic study of access control models by giving a deep description of models, and discussing their reinforcements and weaknesses. we present a design algorithm for access control model that can handle for both existing and unknown users of the database. The algorithm deals with three major parts, Environment Check, Roles and Permissions Check and finally the increment and decrement of permissions dynamically.

Keywords: Dynamic Access Control; MAC; DAC; Environment-based; RBAC

1. INTRODUCTION

Nowadays, in order to research, marketing or provide better service, a numbers of enterprises would collect customers' relevance data, such as personal information, service experience, and desired functions. However, since the occurrences of deceptive crime and personal information disclosure happened frequently, privacy protection has been paid much attention by consumers, companies, researchers, and legislators. Victims not only receive annoying advertisements and reluctant marketing tricks, but also face to the threat of life and property (Babu,et al. 2013. Therefore, the enterprises, government and data providers need to raise privacy-aware consciousness. To raise privacy aware consciousness, data providers should take notice of the private level of delivered data and ensure the transmission security, content confidentiality and supplementary measures like contracts establishment (Peng et. al 2008). If the mechanism for privacy protection is defective, we should prefer rejecting to provide sensitive data to indiscriminately exposing private information which may result in facing the threat of life and property. Fortunately, businesses gradually have built up customer dependence by practicing privacy protection mechanism, consequently, they avoid losing potential profits and attract more customers as much as possible (Barker and Stuckey, 2003). The traditional access control models are discretionary access control (DAC) and mandatory access control (MAC).

These traditional models have shortcomings when dealing with complex, large systems, with possibly hundreds of users and thousands of data items. In DAC models, permissions are assigned to subjects directly. The disadvantage of such an approach is that, in a very large system, the granting of permissions to operate on individual data items to individual users is very time consuming and difficult to manage. It is also difficult to remember which permissions should be revoked from users when they leave the company or change jobs. MAC models, on the other hand, are very rigid, requiring that a lattice-based set of labels be applied to all objects and subjects and that constraints concerning reading and writing of objects must be satisfied (Sandhu,1993). MAC models are designed for applications where the keeping of secrets and the control of information flow are the primary requirements. It is very difficult to design a commercial security model which has such strict constraints on information flow. Role-based access control (RBAC) models have been discussed since the mid 1990s (Nyanchama and Osborn , 1994). RBAC (Sandhu et. al. 1996) is either a NIST standard (Ferraiolo et al 2001) or an alternative measure of both MAC and DAC to directly aid function-based and job-based access control. The main goal of RBAC systems is to provide a model and tools to help manage access control in a complex environment with a very large number of users and an even larger number of data items. Since the introduction of RBAC models, many embellishments have been proposed, including administration, delegation and complex constraints. The RBAC model as shown in figure 1 consists of four entities: User, Role, Permission and Session.

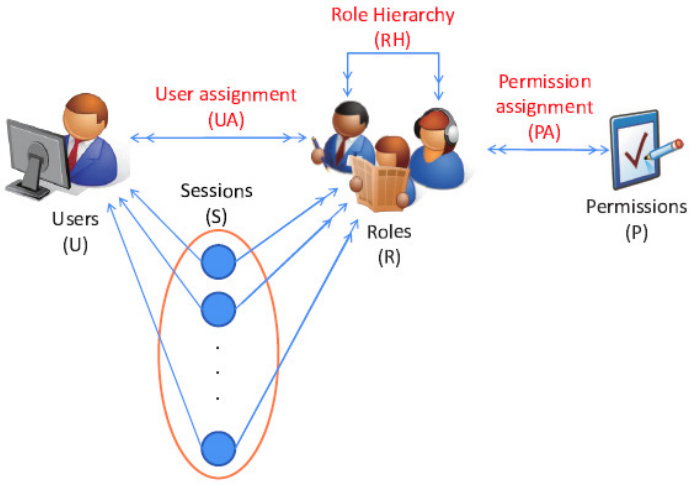


Figure 1: The conceptual models of an original role-based access control

User: User is a human being related to the entire internal and external enterprise system. All users, such as employees, customers, and business partners, have their own position and duty in an enterprise.

Role: Role is a named job title or job function which defines an authority level. If an user has been assigned and authorized a role (User Assignment; UA), he/she can exercise a permission to access specific data. The relations between users and roles are many-to-many, in other word, an user can belong to many roles, and a roles can assigned to many users. Figure 1 shows a special model in the framework, Role hierarchy (RH). Hierarchies are means for structuring the relations between roles to reflect an organization's lines of job function, class, and responsibility. For example, the relation through a superintendent, a primary-care physician, and a health-care provider in a hospital is a hierarchy structure. The senior-most role is that of superintendent, and a health-care provider is junior to a primary-care physician. It means a superintendent could inherit all permissions from primary-care physician and health-care provider because of transitive inheritance.

Permission: A permission can be the terms authorization, access right, and privilege in the literature. Permissions confer on their holder the ability to perform an action such as read, write, and execute in the system. After assigned specific permissions (Permission assignment; PA), a role has rights to do operations designated in permissions to data. Based on many-to-many relation, a role can hold many permissions, and the same permission can be appointed to many permissions. If the data provider want to protect his/her sensitive privacy data, he/she could edit permissions for restricting rights or specific conditions to access data he/she own. For instance, a data provider, Bob can set a permission like "only the salesman can read Bob's personal information for marketing at 9 to 17 o'clock. In this example, Bob sets four restrictions on his personal information, including access operation "read", role assignment "salesman", specific purpose "marketing" ,and time period "9 to 17 o'clock". By this way to protect privacy of data providers, they could rely on their intention to adjusting permissions flexibly.

Session: Users can activate a subset of the roles simultaneously for establishing sessions. A user invoke the roles they belong to enable to accomplish tasks in a session.

RBAC is a wildly used approach to restricting system access to authorized users in computer system security. The permissions which perform certain operations of resources are assigned to specific roles. It means that RBAC regulates users' access based on rights and authorization of their roles. One of the reasons why we adopt RABC to protect privacy is its authorization management mechanism. RBAC is designed to meet the need of relieving the authorization management and immediately offering access control policies (Peng et. al 2008). Therefore, more and more companies adopt the RBAC model for access and authority control using many commercial products (e.g. ORACLE database system, ORACLE IDM, IBM Tivoli IDM, and Sun IDM) and support services (e.g. Role Engineering and Role Mining) (Bertino et. al. 2010). The permission is assigned by a data provider, and only authentic users playing the authorized role could access data. Because of the security policies of the organization and ability of privacy protection by using purpose restrictions, many researches combine these two fields to achieve the objective of privacy protection.

2. RELATED WORK

In (Bertino et al., 2010) implement the notion of role-based access control for preserving private information by properly structured restriction. In (Hung, 2005), the confidentiality of personal identifiable information and protected health information is important for patients. The author presents a framework of RBAC with privacy-based extensions in e-Healthcare services. In (Peleg, 2008), in order to practice in controlling access to sensitive data, such as electronic health record, the authors develop a Situation-Based Access Control (SitBAC) model. It not only protecting patients' privacy but also regulate the concerning data access used by employees. The above mechanisms devote to protect users' privacy in a particular environment by constructing an privacy-aware system using the concept of integration of the model of role-based access control. In (Bertino et al., 2010), the authors propose a comprehensive frame work applying a family of privacy-aware role-based access control (P-RBAC) as Figure 2 to enforce access control to data containing personal or sensitive privacy information. A family of role-based access control models is the key feature that extends classical RBAC on taking into account purposes and obligations. There is four models in the family: core, hierarchical, conditional, and universal P-RBAC. Core P-RBAC models includes five basic elements: Users, Roles, Objects, Operations, and Permissions as via 4. Core P-RBAC is based on Core RBAC without the session component. Hierarchical P-RBAC models and Conditional P-RBAC models extend Core P-RBAC with advantages of additional components to be appropriate for various requirements of different enterprises. In (Hung, 2005), the entity "User" refers to persons who may use the e-Healthcare service system including doctors, patients, administrators, and insurance companies, etc., and "Role" is the job title or job function of an user. "Object" refers to attributes (e.g., salary, age, and department etc.) related to users or something essential (e.g. e-mail) in processes of data access control. After defining entities above in advance, if an user is about to do "Operation"(e.g., read/write) on privacy data such as protected health information, the "Permission" set up by data owners define restrictions such as specific role, object, condition, and obligation, etc..

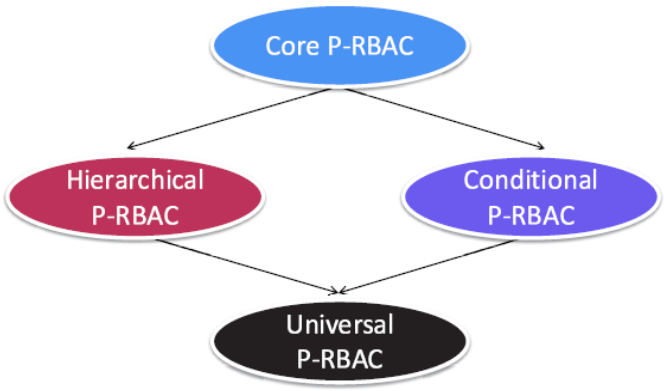


Figure 2: The family of conceptual P-RBAC models

Hierarchical P-RBAC models contain Role Hierarchical (RH), Object Hierarchical (OH), and Purpose Hierarchical (PH). We survey some papers of hierarchical applications, such as (Cai et al., 2009) defined clearly about the role relationships, abuse inheritance, and security principles. The authors of (Byun and Li, 2008) use the concept of hierarchical purpose to protect the privacy of users. The concept of a hierarchical structure is an order relation between two different individuals. For instance, given two roles $R1 \Subset R2$, if a policy defines the data access authority to $R1$, $R2$ could inherit this permission to get the data access authority because of a higher role level than $R1$. Conditional P-RBAC models (Ni et al., 2007) provide permission assignment sets and complex Boolean expressions. Conditional P-RBAC models support not only new context variable types like string and integer, but also logical operators like negation and disjunction.

Universal P-RBAC combines the character of either Hierarchical P-RBAC or Conditional P-RBAC, and provides additional three features, negative permissions, flow control for obligation execution, and permission combination principles.

Thus, after historical development, researchers combine role-based access control with privacy protection mechanism to construct the family of four models for supporting demands components strongly

3. ACCESS CONTROLS MODELS

*(a)**The DAC model (Discretionary Access Control)***
The Discretionary Access Control model (DAC) (Lampson, 1974) allows a subject to assign permissions to other subjects. This access control is flexible, but it can cause errors. The agreement or revocation of privileges is regulated by an administrative policy. The management access to the files of the operating system UNIX is a classic example of access control mechanism based on a discretionary policy. We will present in the following the two well known discretionary models, that are the Lampson model and HRU model (Harrison Ruzzo Ullman model).

*1) **The Lampson model**:* The concept of access rights specified by a matrix of access control was introduced in 1971 by Lampson. This model is represented by a triple (S, O, M), where S denotes the subjects, objects O and M= (Mso) the access control matrix that associates to each couple (subject s, object o) a set of access rights that are usually: read, write, run.

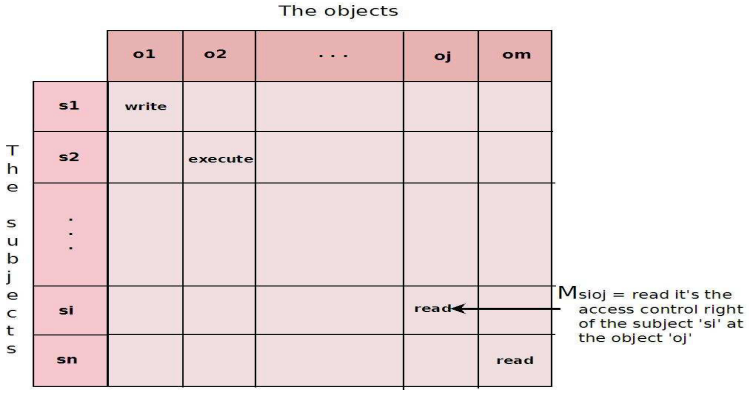


Fig. 3. Matrix of access control

The matrix shown in Figure 3 shows that the right of access is associated with the subject si and the object oj . While the matrix is not fixed yet, it can be updated by the creation of new objects or subjects, by the destruction of the latter as well as the addition or removal of access rights.

***2) The HRU model**:* The Harrison Ruzzo Ullman model (HRU), formalized in (Harrison, 1976) represents an improvement of Lampson model. This model uses a classical access matrix like the Lampson model. The difference lies in that HRU specifies the commands (a set of primitive operations) to assign access rights (read, write, own, etc.), as well as create and delete subjects and objects. In this model, if the right **"own"** is associated with a pair **(s, o)**, the subject s will be considered as the owner of the object o and it may assign its rights of access on the object o to other subjects. In other words, this action allows the subject to define the permissions on the entire column. The possible primitive operations are: **Enter**: for adding rights, **Delete**: Delete rights, **Create subject**: to create new subjects, **Create object**: to create new objects, **Destroy subject**: the destruction of subjects and **Destroy object**: the destruction of objects. The commands in the HRU model are built from primitive operations above and take as argument subjects and objects. We can add a right r in an access matrix M_{so} if there is a command C that adds the right r in a cell of Mso: $c : M_{so} \rightarrow M'_{so}$ i.e. $\exists_s, o : r \notin M_{so} \ \Lambda \ r \in M'_{so}$

The HRU command has an optional conditional part as well a body, it has the following format:

command $c(x_1, ..., x_k)$
if a_1 in $M_{s1,o1}$
...
a_n in $M_{sm,om}$
then op_1
...
op_n
end

With n>0, m≥0, a_1, ..., a_n are authorizations, op_1, ..., op_n are primitive operations. HRU command may not have condition (m=0). We note that despite the fact that we trust users so they follow the policies of the organization, we can not trust the processes running on their behalf, hence the need to distinguish between users and processes that are running for their accounts (subjects).

*B). **The MAC model (Mandatory Access Control)***
The Mandatory Access Control model (MAC) (Bell and Padula, 1976) allows to create obligatory security policies that set essential rules to force compliance of access control requirements. Thus, unlike the DAC model, users can not define the rights of access control, because all objects are the exclusive property of the organization, which implies that in this model the access control policy is managed in a centralized manner. The Mandatory Access Control (Sandhu, 1973) is based on the concept of security levels associated with each subject and object, from which are derived the permissions and actions. A mandatory policy of security, is only a multi-level policy (Denning ,1976), this latter has the notion of access class. A partial order relation is defined on the set of access classes, it is the dominance relation symbolized by ≥. Each access class has two component:

• **Security Level**: is an element of a totally ordered set, e.g. top secret (TS), secret (S), confidential (C) and unclassified (N) where TS ≥ S ≥ C ≥ N. For objects, security level is called the classification level and for subjects it is called clearance level.

• **A set of categories**: describes the various fields of system in study. For example, in military systems, the categories are nuclear and army, in commercial systems the categories are rather financial, administrative...

Let L be the set of security levels, equipped with the partial order relation ≥ , and C is the set of categories, equipped with the partial order ⊇. Let l_1, l_2, two levels and c_1, c_2 two categories such as: $l_1 \in L$, $l_2 \in L$, $c_1 \in C$, $c_2 \in C$. Given two access classes ac_1 and ac_2, the dominance relation ≥ is defined as:

$\forall ac_1 = (l_1, c_1), ac_2 = (l_2, c_2) : ac_1 \geq ac_2 \iff l_1 \geq l_2 \wedge c_1 \supseteq c_2$

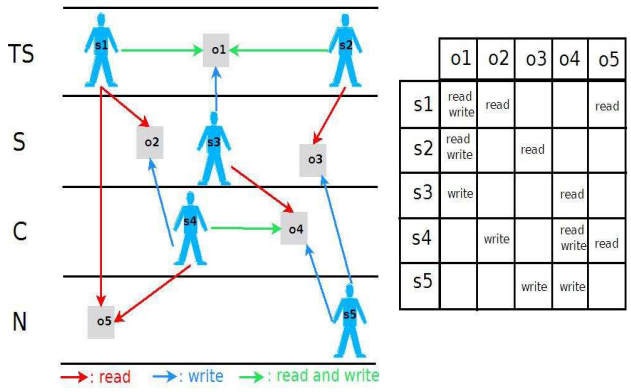
The structure of all access classes forms a trellis that is why multi-level policies are also called by LBAC (Lattice based access control). To summarize, the multilevel model with all its variations is based on the trellis concept, it also uses an access matrix identical to the HRU model in order to present authorizations on which are added security levels. We will then speak of the two most famous models of Mandatory Access Control, that are the Bell-LaPadula model (BLP) which has the purpose to ensure confidentiality, and the Biba model which is interested to integrity.

*1) **The Bell-LaPadula model (BLP)**:*
The Bell-LaPadula model (BLP), developed in (Bell and LaPadula, 1975) seeks to preserve the confidentiality of the data, that is to say that these latter are only accessible by authorized users see figure 4. In this model, access rights depend classifications assigned to objects and authorizations granted to subjects, basing on two laws:

a) **Simple property (no read up)**: simply do not read up. In effect, this law prohibits a subject to have a read access to an object that has a higher classification than the habilitation of the same subject:

$read \in M_{so} \Rightarrow f(s) \geq f(o). (f : S \cup O \rightarrow L)$

b) **Star property (no write down)**: simply do not write down (write is used to mean the only writing or addition). In effect, this law prohibits a subject to have a write access to an object that has a classification lower than the habilitation of the same subject:
$write \in M_{so} \Rightarrow f(o) \geq f(s)$.



	o1	o2	o3	o4	o5
s1	read write	read			read
s2	read write		read		
s3	write			read	
s4		write		read write	read
s5			write	write	

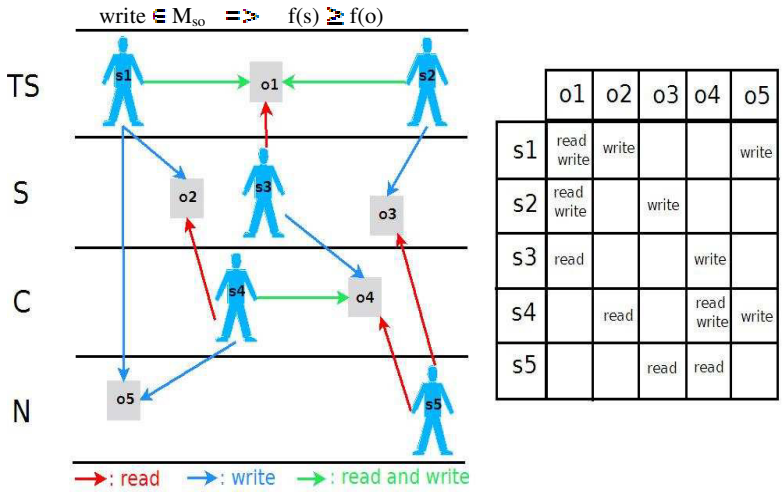
Figure 4. Two laws of BLP model

2) The Biba model: The Biba model developed in (Landwehr ,1981) focuses on data integrity, what is missing in the BLP mode. Ensuring data integrity means that they can only be changed by authorized users see figure 5. As in BLP, the Biba model is based on two laws:

a) **Simple property (no read down)**: simply do not read down. In effect, this law prohibits a subject to have a read access to an object that has a classification lower than the habilitation of the same subject:

read $\in M_{so}$ $\Rightarrow$ f(o) $\geq$ f(s)

b) **Star property (no write up)**: simply do not write up (write is used to mean the only writing or addition). In effect, this law prohibits a subject to have a write access to an object that has a higher classification than the habilitation of the same subject:



	o1	o2	o3	o4	o5
s1	read write	write			write
s2	read write		write		
s3	read			write	
s4		read		read write	write
s5			read	read	

Figure 5. Two laws of Biba model

4. FORMAL DESCRIPTION OF RBAC

To clarify the notions presented in the previous section, we give a simple formal description, in terms of sets and relations, of role based access control. For each subject, the active role is the one that the subject is currently using:

AR(*s*: *subject*) = {*the active role for subject s*}.

Each subject may be authorized to perform one or more roles:
RA(*s*: *subject*) = {*authorized roles for subject s*}.

Each role may be authorized to perform one or more transactions:

TA(*r*: *role*) = {*transactions authorized for role r*}.

Subjects may execute transactions. The predicate *exec(s,t)* is true if subject *s* can execute transaction *t* at the current time, otherwise it is false:
exec(*s*: *subject*, *t*: *tran*) = *true iff subject s can execute transaction t.*
Three basic rules are required:

Role assignment
A subject can execute a transaction only if the subject has selected or been assigned a role:

$s : subject, t : tran, (exec(s,t) \Rightarrow AR(s) \neq \emptyset)$.

The identification and authentication process (e.g. login) is not considered a transaction. All other user activities on the system are conducted through transactions. Thus all active users are required to have some active role.

Role authorization
A subject's active role must be authorized for the subject:

$\forall s : subject, (AR(s) \subseteq RA(s))$.

With (1) above, this rule ensures that users can take on only roles for which they are authorized.

Transaction authorization
A subject can execute a transaction only if the transaction is authorized for the subject's active role: $\forall s : subject, t : tran, (exec(s, t) \Rightarrow t \in TA(AR(s)))$.

With (1) and (2), this rule ensures that users can execute only transactions for which they are authorized. Note that, because the conditional is ``only if'', this rule allows the possibility that additional restrictions may be placed on transaction execution. That is, the rule does not guarantee a transaction to be executable just because it is in *TA(AR(s))*, the set of transactions potentially executable by the subject's active role. For example, a trainee for a supervisory role may be assigned the role of ``Supervisor'', but have restrictions applied to his or her user role that limit accessible transactions to a subset of those normally allowed for the Supervisor role.

5.. ENVIRONMENT BASED DYNAMIC ACCESS CONTROL MODEL (EBDACM)

Environment based Dynamic Access Control Model (EBDACM) addresses the dynamic access control requirements when an unknown user or registered user wants to access the database of an organization. It is a different model and an extension to the existing RBAC (Ahmad, 2011). This model deals with the following requirements.

1. Appropriate assessment of a user according to his environment and assigning the permissions according to his/her role.
2. The permissions assigned to him/her are dynamically changed based on his/her access pattern.
3. The permissions can dynamically be incremented or decremented after analyzing the access pattern of the user after interval of every n minute.

All of this need to be done when the organizations keep their database as open source and different users belonging to different organizations tends to access the data. So being open source, data needs to be protected from being accessed by unauthorized users. Secondly this model is flexible enough to increase or decrease the access permissions of a user based on the security checks. Figure 6 below shows the conceptual framework of the model. Basically, the environment checking will be done at two places. Once, at the beginning access stage when a user tries to connect to the database. The environment factors that are going to be checked will be among others the origin of the user, i.e., the URL from which the user accesses the database and the software used by the user. After the user has passed the initial check and given the appropriate initial access privileges which are determined by referring to the RBAC storage, he/she can start using the database. Then, after some time interval, if he/she is still attached to the database, his/her access pattern will be evaluated. The access pattern that the system still will look for will be in terms of security threats.

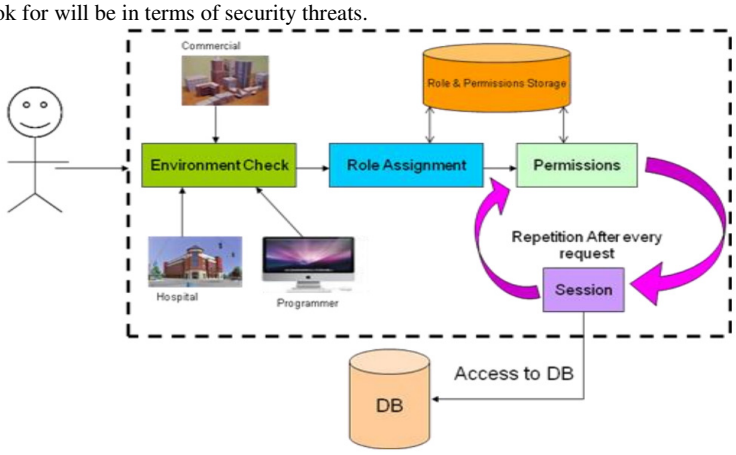


Figure 6. Environment Based Access Control Model

EBDACM addresses the security threats which are described in Table 1 below. All the threats that are highlighted in the table refer to the access control of database. If the user is using the system without any harm or threat to the data, then possibly his/her access privileges will be increased.

Table 1. Security Threats Addressed by EBDACM

Source	Threats	
Britt [13]	Access Control Data Compromised	Unauthorized Access
Scheier [14]	Access Control	Data Sensitivity
Schultz [15]	Access Control • Data Breaches	Unauthorized Access

If the user is caught while causing some kind of harm or threat to the database, his/her access permissions may be decreased or terminated altogether. Again both the increment and decrement of access privileges will be referred to the RBAC storage as in Table 2.

Table 2. RBAC Storage Structure

Role	Permission (P)		
	Level1	Level2	Level3
XX	{Table1, Table2}	{Table3}	{Table4}
YY	{Table4, Table6}	{Table1, Table2, Table3}	
UNKNOW	{Table1}	{Table2}	{Table3}

Like a normal RBAC, privileges will be assigned based on roles of users in the organization of the database. For example, there are two types of users at the organization, XXX and YYY. The users, if verified, will be assigned privilege of level 1. If after certain time interval, the access pattern of the user does not show any threat, the user's privilege will be increased to include privilege of Level 2. Similarly, if the user has exhibited some threats, his/her access level will be demoted or terminated if he/she is already at level 1. The same concepts apply to unknown users.

6. ALGORITHM

The following are the algorithm and the notations used in the model.
NOTATIONS
U_{new} is the totally new user attempting to access ; U_e is the existing user : R_c is the role check
Θ is the value for access pattern : SW_i is the software check to examine the interface
P is the permissions given to users : O_p is the operation performed

Step 1: Environment Check
User requests to login
Open Log File
Check the user's log information
IF(U_{new})
Begin
IF(SWi = 1)
Update log file
ELSE
Display ―can't give permissions‖
Exit
End IF
ELSE
Go to next Step

The working of our model can be summarized as follows.
First of all our system will check the user's environment that from which place he/she is trying to connect with the database. Does the user belong to an organization that is already connected to the database or he/she is totally new user. On the second phase the model will check for the interface of the user that he/she is using to connect with the database. This is done just to make sure that the connecting PC the software is completely clean of viruses.

Step 2: Role Check & Permissions

IF(U_e)
See log file (for previous permissions record)
P = P + 1
Update log file
ELSE
P = P + 1 (Only view permission)
Update log file
END IF
Go to next step

At the third stage the system will check the access pattern of the user based on which it will allow him/her the permissions of accessing the database. The dynamicity in the function, (that is the actual concern of the model) the system will continuously check after each n minutes interval, the access pattern of the user to permit him/her further privileges. If the user is using the system quite safely then the model allows the user to get further permission and if not then it has the ability to eliminate the user from the access range.

Step 3: Permissions Increment or Decrement

User requests to access New Table or Alter Data
See log information for user
IF (Ue)
IF (R = 1)
P = P + 1
Routine Called
Table Backup
ELSE
Access Denied
IF (Op = L)
Update Database
Update Log file
ELSE
Replace Table with Backup Table
P = P – 1
ELSE
IF (R = 1)
P = P + 1
Routine Called
Table Backup
ELSE
Access Denied
IF (Op = L)
Update Database
Update Log file
ELSE
Replace Table with Backup Table
P = P – 1
END IF

7. CONCLUSION

Access control is a very important area in the security of information systems because it ensures the confidentiality and integrity of data. This is what motivated us to do a study about the most famous access control models. As we have seen in this paper, we have tried to talk about two types of access control models, the Discretionary and Mandatory models, and then we have detailed the RBAC model. This paper also presented the Environment-Based Dynamic Access Control Model (EBDACM) that provides the dynamic environment check and assignment of permissions to a user of the database. It extends the existing RBAC model and dynamically adjusts the role assignment and permission assignment based on the environment and the access pattern of the user and automatically decides whether the user should continue its access to database or not. On the other hand it also decides that the user may get its permissions increased based on its safe and sound access pattern. Compared to the traditional access control models EBDACM provides a dynamic and improved security for normal databases in the organizations. For future work, we will refine our algorithm so that the actual steps of checking threats can be done as well as automating our RBAC structure of privilege levels.
.

REFERENCE

[1] Ahmad S, Ahmad R, ―Environment-based Dynamic Access Control Model for Database Systems‖ 2011 3rd International Conference on Machine Learning and Computing (ICMLC 2011)) 978-1-4244-925 3-4 /1
[2] B. S. Babu and N. Jayashree and P. Venkataram, "Performance analysis of Steiner tree-based decentralization mechanism (STDM) for privacy protection in wireless sensor networks," *International Journal of Network Security*, vol. 15, no. 5, pp. 321-330, 2013.
[3] S. Barker and P. J. Stuckey, "Flexible access control policy specification with constraint logic programming," *ACM Transactions on Information and System Security*, vol. 6, no. 4, pp. 501-546, 2003.
[4] E. D. Bell and J. L. La Padula, "Secure computer system: Unified exposition and multics interpretation," Bedford, MA, 1976. [Online]. Available: http://csrc.nist.gov/publications/history/bell76.pdf
[5] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," pp. 74–244, 1973.
[6] J. W. Byun and N. Li, "Purpose based access control for privacy protection in relational database systems," *Information Systems Frontiers*, vol. 17, no. 4,pp. 603-619, 2008.
[7] W. Cai, R. Huang, X. Hou, G. Wei, S. Xiao, and Y. Chen, "Atom-role-based access control model," *IEICE Transactions on Information and Systems*, vol. E95.D, no. 7, pp. 1908-1917, 2012.
[8] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, no. 5, pp. 236–243, May 1976. [Online]. Available: http://doi.acm.org/10.1145/360051.360056
[9] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed standard for role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 224-274, 2001.
[10] P. C. K. Hung, "Towards a privacy access control model for e-healthcare services," in *Third Annual Conference on Privacy, Security and Trust*, Oct.2005.
[11] Gallaher, M., A. O'Connor, and B. Kropp. 2002. *The Economic Impact of Role-Based Access Control*. Prepared for the National Institute of Standards and Technology. Research Triangle Park, NC: RTI International.
[12] W. S. Juang and J. L. Wu, "Efficient user authentication and key agreement with user privacy protection," *International Journal of Network Security*, vol. 7, no. 1, pp. 120-129, 2008.
[13] B. W. Lampson, "Protection," *SIGOPS Oper. Syst. Rev.*,vol. 8, no. 1, pp. 18–24, Jan. 1974. [Online]. Available: http://doi.acm.org/10.1145/775265.775268
[14] C. E. Landwehr, "Formal models for computer security," *ACM Comput. Surv.*, vol. 13, no. 3, pp. 247–278, Sep. 1981. [Online]. Available: http://doi.acm.org/10.1145/356850.356852
[15] Kuhn, D.R., E.J. Coyne, and T.R. Weil. 2010. "Adding Attributes to Role-Based Access Control." *Computer* 43(6):79-81.
[16] M. Nyanchama and S. L. Osborn. Access rights administration in role-based security systems. In J. Biskup, M. Morgenstern, and C. E. Landwehr, editors, *Database Security, VIII, Status and Prospects*, pages 37–56. North-Holland,1994.
[17] Q. Ni, E. Bertino, J. Lobo, C. Brodie, C. Karat,J. Karat, and A. Trombetta, "Privacy-aware role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 3,July 2010.
[18] H. C. Peng, J. Gu, and X. Ye, "Dynamic purpose-based access control," in *Parallel and Distributed Processing with Applications, 2008. ISPA '08. International Symposium on*, pp. 695-700, 10-12 Dec. 2008.
[19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer archive*, vol. 29, no. 2, pp. 38-47, 1996.
[20] R. Sandhu. Lattice-based access control models. *IEEE Computer*, 26:9–19, Nov.1993.
[21] R. S. Sandhu, "Lattice-based access control models," *Computer*, vol. 26, no. 11, pp. 9–19, Nov. 1993. [Online]. Available:http://dx.doi.org/10.1109/2.241422
[22] Q. Ni, D. Lin, E. Bertino, and J. Lobo, "Conditional privacy-aware role-based access control," in *Proceedings of the European Symposium on Research in Computer Security*, Springer-Verlag, Berlin, pp. 72-89, 2007.
[23] M. Peleg, D. Beimel, D. Dorib, and Y. Denekamp,"Situation-based access control: Privacy management via modeling of patient data access scenarios," *Journal of Biomedical Informatics*, vol. 41, pp. 1028-1040, Dec. 2008
[24] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Commun. ACM*, vol. 19, no. 8, pp. 461–471, Aug. 1976. [Online]. Available: http://doi.acm.org/10.1145/360303.360333