Design of Algorithm for Environment based Dynamic Access Control Model for Database Systems

Sajjad Ahmad
Department of Computer &
Information Sciences
Universiti Teknologi PETRONAS
Perak, Malaysia

Rohiza Ahmad
Department of Computer and
Information Sciences
Universiti Teknologi PETRONAS
Perak, Malaysia

ABSTRACT

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, 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

Database; Access Control; Dynamic Access Control; Environment-based, RBAC

1. INTRODUCTION

Access control is the conventional hub of importance of computer security. It is where security engineering tends to meet computer science. Its purpose is to manage which major associates have access to which part in the system—which files they have the access to read, which programs they can run, how they share data with others so on and so forth.

The access control methods, which the user can have at the application level, have the ability to express a very prosperous and composite security strategy [1]. A contemporary online business might allot staff to one of dozens of multiple roles, and every one of them can possibly start some subset of quite a few hundred potential dealings in the respective system. Some of these may need online approval from a third party while others may want double control [2].

The applications of access control might be written on peak of middleware when we talk about the hardware and operating system acquaintance, just like a database management system that imposes a numerous safety properties [3]. The middleware will obviously use services offered by the fundamental operating system. Because this creates resources like files and communications ports from lower-level components, it obtains the liability for as long as ways to organize access to them [4]. Furthermore, Access control is apprehensive with shaping the permissible actions of legal users; mediate each effort by a user to access a part of data in the system [5]. A given information technology (IT) mechanism may apply access control systems in various places and at different levels. Operating systems mostly use access control to guard records and information bank. Database management systems DBMS may apply access control to control access to tables and views. Most business related accessible application systems put access control into operation, repeatedly autonomous and mostly independent of the operating systems and/or DBMSs on which they are installed [6].

The purpose of an access control system is often described in terms of protecting system resources in opposition to unsuitable user access. From a business point of view, this objective can just as well be explained in terms of the best distribution of information. Above all, the major goal of IT is to make information accessible to users and applications. A greater degree of sharing may become a hurdle in resource protection; in actuality, a well-organized and useful access control system actually facilitates distribution of information [7]. An adequately classy access control system can facilitate careful sharing of information where in its absence, sharing may be measured so much unsafe in total.

Hence, this paper will present section 1.1 Important Notations used for this model, section 1.2 Role based Access control model, related work in section 2, the proposed model in section 3, proposed design in section 4, the algorithm in section 5, followed by discussion and conclusion in section 6 and 7 respectively.

1.1 Notations used in Access Control System

Some of the notations used are discussed as follow.

- **Object:** An entity that holds or is given information. Access to an object very oftenly entails access to the information it has. Examples of objects are records, fields (in a database record), blocks, pages, segments, files, directories, directory trees, process, and programs, as well as processors, video displays, keyboards. Devices like electrical switches, disc drives, and associated to a computer system can also be integrated in the group of objects.

- **Subject**: A lively unit, usually in the form of a person, process, or device that has the source of information to pour among objects or changes the system position.

- **Operation**: An active process called upon by a subject; for example, when an automatic teller machine (ATM) user enters correct personal identification number (PIN), the control program operation on the user's behalf is a procedure, but the subject can begin more than one operation-deposit, withdrawal, balance inquiry.

- **Permission (privilege):** An authorization to carry out some operation on the system. The term permission refers to some mixture of object and operation in computer security literature. A specific function used on two different objects represents two separate permissions, and likewise, two different operations applied to a sole object signify two different permissions.

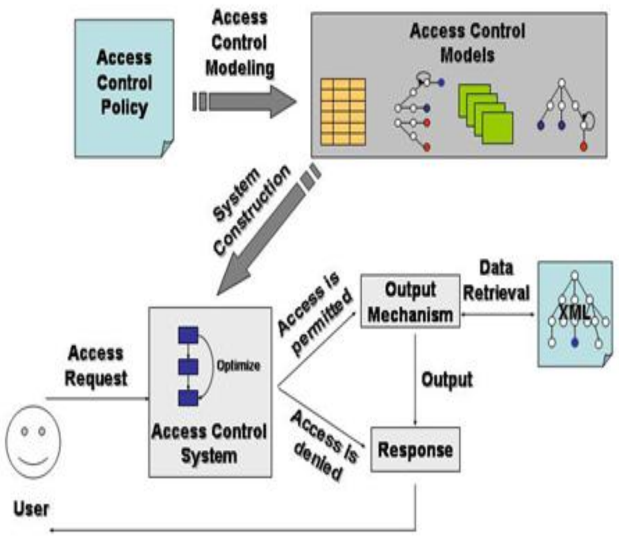

Fig 1. System Flow Diagram of Access Control Model (ACM)[8]

- **Access Control List (ACL):** A list connected with an object that identifies all the subjects that can approach the object, alongside with their privileges to the object. Each entry in the list is a couple (subject, set of rights). An ACL communicates to a file of the access control matrix. ACLs are regularly executed openly or as an approximation in contemporary operating systems.

1.2 Role Based Access Control Model (RBAC)

Within an organization, roles are created for a variety of job purposes. The permissions to carry out definite operations are assigned to precise roles. Members of staff (or other system users) are assigned specific roles [8], and through those role assignments obtain the permissions to execute fastidious system operations. Because the users are not assigned permissions openly, but only get them through their role (or roles), management of every single user privileges becomes a substance of simply handing over suitable roles to the user; this makes general operations more easy, such as adding a user, or changing a user's department [8][9].

Three primary rules are defined for RBAC:

1. Role assignment: A subject can carry out an operation only if the subject has selected or been assigned a role.
2. Role authorization: A subject's current role must be endorsed for the subject. With rule 1 above, this rule makes certain that users may get on only roles for which they are authenticated.
3. Transaction authorization: A subject can perform an operation only if the operation is approved for the subject's active role. With rules 1 and 2, this rule guarantees that users can perform only operations for which they are authorized.

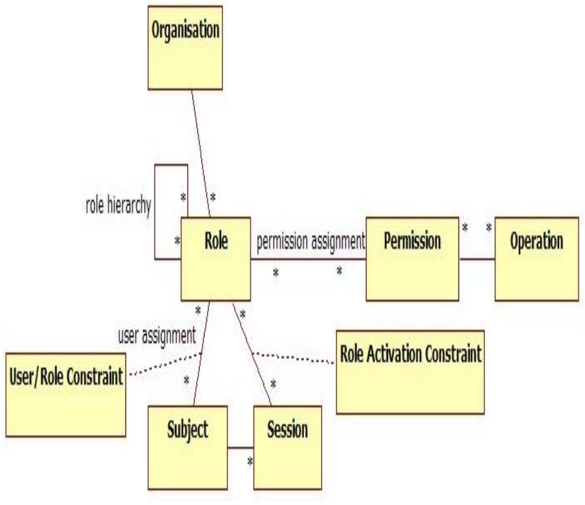

Fig 2. System Flow Diagram of Role Based Access Control Model (RBAC) [8]

Extra limitations may be applied as well, and roles can be combined in a chain of command where higher-level roles include privileges possessed by sub-roles.

With the emergence of business trend that requires organizations to go borderless, the protection of an organization's private data has become a major focus of current research areas regarding information security. Privacy is a term that can be defined as the personal right of an individual or an organization to decide itself on when, how, to whom and to what extent it will be sharing its private information [10]. In conventional access control models, it has been very rarely seen that the proper privacy policies are enforced [11], especially based on the environment in which the user belongs to. This is because the working of the traditional access control models is based on a few operations of the users on a specific data [12] rather than the general access with specified permissions to data.

For the current popular use of Role Based Access Control (RBAC) model [13] in assigning access level to users, we may consider that it is very much possible to extend the model so that it recognizes or supports privacy protection by evaluating the users' purposes, conditions and obligations when using a database. Most of the database organizations have integrated RBAC model into their database management systems, and the privacy-aware models framed by enhancing RBAC model can be usefully implemented into these systems [14]. The mandatory implementation of privacy policies must be assured; otherwise organizations may intentionally or unintentionally break their privacy policies in the real time performance.

2. RELATED WORK

Currently A Dynamic Access Control model Based on Trust (DACBT) [15], Purpose Based Access Control Model for Privacy Protection in Relational Database Systems (PBAC) [16], Dynamic Purpose-Based Access Control (DPBAC) [11], Domain-Based Access Control for Collaborative E-Commerce System (DBAC) [17], A Context-Aware Access Control Model for Pervasive Computing in Enterprise Environment (CTRBAC) [18] and Actor and Trust-Based Dynamic Access Control Model in Universal Computing Environment (ATBAC) [19] are observed as the most important information privacy preserving models.

The DACBT model highlights that when a user puts on a request to access the information, the decision of sanction of privileges depends on the role of the user and trust control. Trust control specifies the faith and ability of a user to have credible and safe operations. Trust values are also defined as the extent of trust, a user's trust in compared with another user and the based on the comparative values, permissions are assigned.

PBAC model on the other hand, assigns access privileges based on user purpose. The model constructs the purpose hierarchy on anticipated purposes and access purposes. Anticipated purposes specify the planned usage of the information and access purposes specify the purposes for which data elements are requested.

The DPBAC model further extends the idea of PBAC by determining the access purpose in dynamic manner rather than static as in PBAC. Access purposes are determined based on the subject attributes, context attributes and authorization policies. As this model is the extension of PBAC, so it introduces the dynamically connected anticipated purposes with the requested data objects during the access privilege's decision making.

DBAC improves the conventional role-based access control model by identifying the idea of domains. Domains allow the design of fine-granularity access control principles in a multilevel method. This reflects the concept to have objects on a common level. For example, an object is a record of database table and every object belongs to a certain domain and the object's domain sometimes may need to change dynamically. The role is restricted to a domain and getting permission is the right of every role.

CTRBAC model integrates the business and pervasive computing requirements by enhancing the Task-Role based Access model dynamic context information. This model highlights three requirements:

1. Suitable roles are turned on according to user's existing environment.
2. Dynamically permissions are assigned to a task.
3. Incompatible bodies in organization should be carefully measured to produce work catalog.

ATBAC model introduces the object of of actor and the dynamic trust model, expanding the RBAC model. It proposes the actor and trust based dynamic access control model. It carries on seal management to the users and the roles of the actors/user. It produces a model from the actor and context perspective. This indicates the permission management dynamically.

RBAC is extensively supported by the dynamic changes in its operations [20][21]. The removal of redundancy and its operations can be implemented very easily on the conflicting factors.

In a location based access control model the access pattern of a user determined based on its location and the distance from the actual location of the data [22]. This model is purely based on mandatory access control model, which highlights the mandatory requirements for an access control.

For pervasive computing environments, Matthew J. Moyer and Mustaque Ahamad [23] have proposed the Generalized Role Based Access Control (GRBAC) model. GRBAC is an extension of traditional RBAC model and enhances it by incorporating the notion of object roles and environment roles, with the traditional notion of subject roles. By defining these three types of roles, i.e., Subject roles, Environment roles, and Object roles, GRBAC uses context information as a factor in making access decisions. But the definition of object roles violates the notion of permission in RBAC and the abstraction of operations on objects could not be achieved. In addition, this problem violates user/role and role/permission associations. So, S. Park et al [24] have proposed an improved model entitled Context-Role Based Access Control (CRBAC) model retaining the notion of permission in RBAC and adds a new notion of Context-role which is similar to Environment roles in GRBAC and represents environment state of the system by a mapping context-roles and context information. By using the uniform notion of a role to capture both user and context attributes, they will build an access control model that grants and applies permissions to users according to current context information. However, both GRBAC and CRBAC may not be feasible in practice because the potential large amount of environment roles or context roles make the system hard to maintain.

In order to perform dynamic access control based on the context information, the Context Agents and State Machines are introduced in the model. There is a Role State Machine for each user and a Permission State Machine for each role. The ole and permission are used as state variables respectively. The Context Agent collects context information and generates pre-defined events to trigger transitions in the State Machines. The idea of dynamic change of user's activated roles and role's activated permissions based on the context information is useful to meet the requirements of pervasive computing and retaining the advantages of RBAC. Context, Rule and Role- Based Access Control (CR-RBAC) [25] Model is also based on this idea. However, it is suitable for pervasive applications such as Aware Home application rather than enterprise environment because the business process logic is not taken into account.

3. ENVIRONMENT BASED DYNAMIC ACCESS CONTROL MODEL (EBDACM)

In [26] we proposed framework for Environment Based Dynamic Access control model for database systems. Detailed description of framework is as under.

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. 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 3 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.

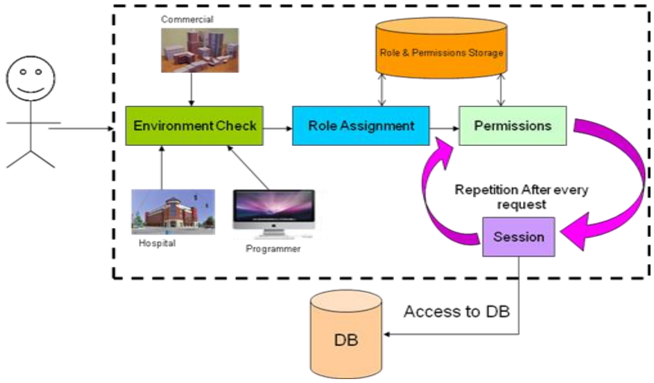

Fig 3. Environment Based Access Control Model

EBDACM addresses the security threats which are described in Table 1 below.

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

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. 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}	
Unknown	{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.

4. PROPOSED DESIGN

In the figure 4 below the proposed design is illustrated. Where the user puts his/her request to enter the database, his/her log information is checked in an existing log file. If the user is found a new user, his/her platform/software interface is checked, if it matches with the database software then the roles and permissions are assigned to users. In the figure 5 below, the next part of proposed design is illustrated. Where permissions are dynamically increased or decreased. User puts the request and system checks its role match and existing permissions. It also checks the log information of the users to make sure that the user is new one or it is an existing user. Then the permissions are increased accordingly. With all this process the system automatically generates a routine that creates a backup table that is kept hidden from user. If the changes that a user makes in a table are not suitable or a user is deleting or modifying some critical data then the system will not allow him to save the data rather than it will over write the changed table with the backup table. If it does so and keeps this operation as a threat and decreases the permissions otherwise it will increase the permissions as the operation was performed smoothly and without any threat to database.

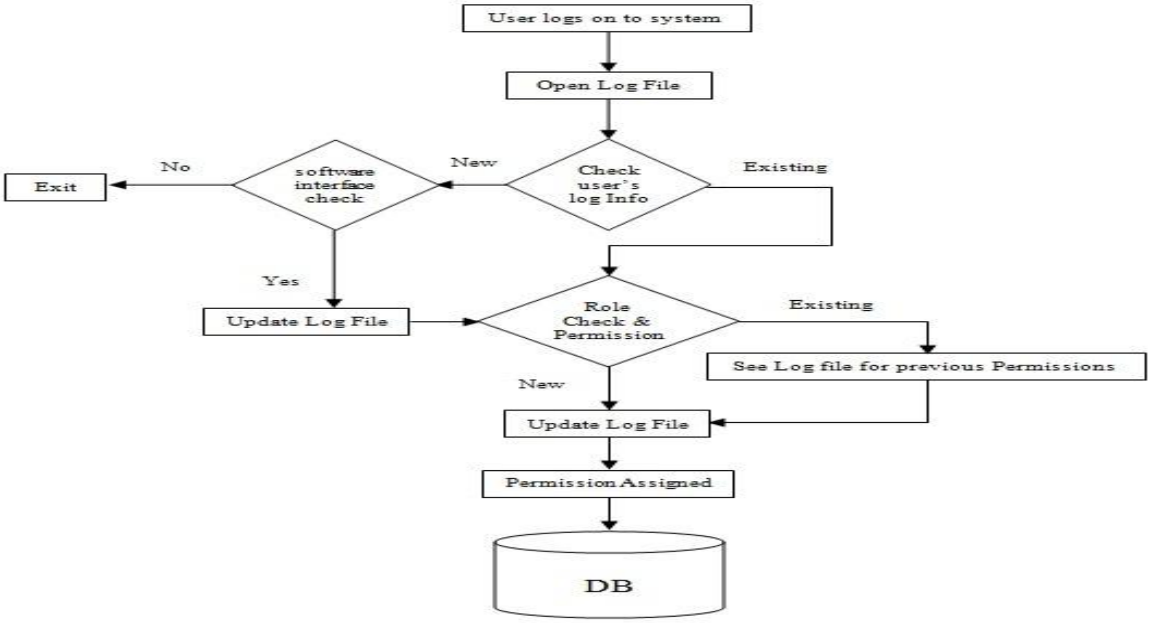

Fig 4. Proposed Design Ist part

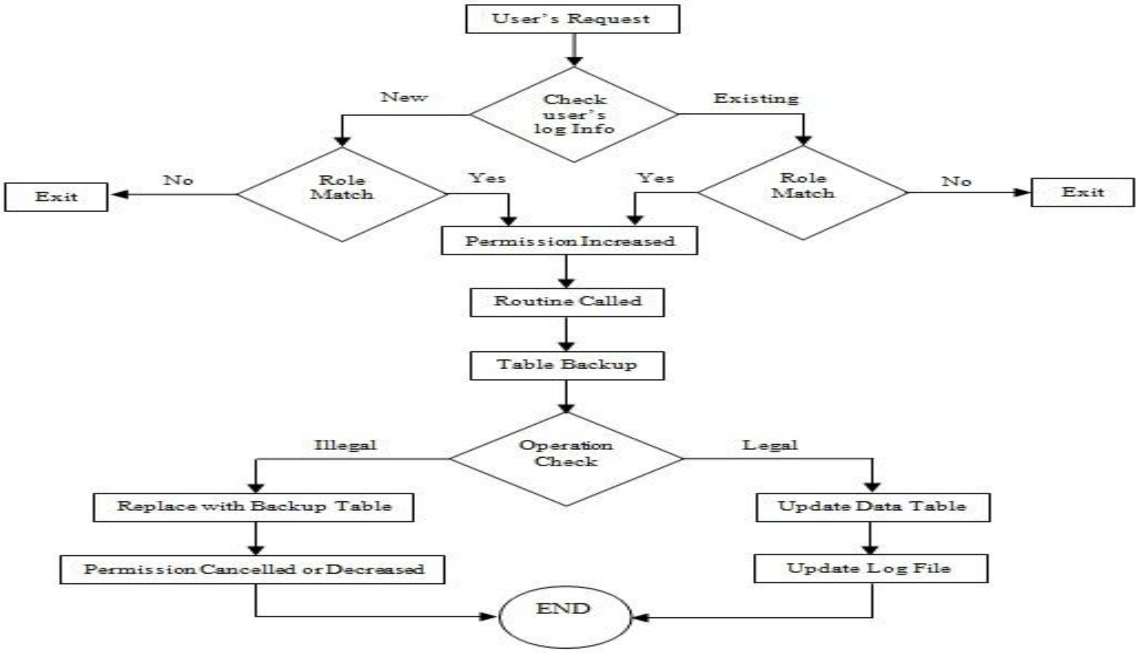

Fig 5. Proposed Design 2nd part

5. ALGORITHM

The following are the algorithm and the notations used in the proposed model.

NOTATIONS

U_{new} is the totally new user attempting to access

U_e is the existing user

SW_i is the software check to examine the interface

R_c is the role check

P is the permissions given to users

O_p is the operation performed

Θ is the value for access pattern

Step 1: Environment Check

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

The working of our proposed 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 ($U_e$)
                IF ($R = 1$)
                        $P = P + 1$
                        Routine Called
                        Table Backup
                ELSE
                        Access Denied
                          IF ($O_p = L$)
                             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 ($O_p = L$)
                             Update Database
                             Update Log file
                          ELSE
                Replace Table with Backup Table
                             $P = P – 1$
        END IF
```

6. DISCUSSIONS

As shown in figure 1 above the EBDACM is the collection of environmental properties of the user. Once it finds the environment and the interface used by the user very much clear regarding the access control security aspect if assigns the roles and the permissions to him/her to access the database. This model basically addresses three main functions of the users while designing and implementing the system prototype. The first is the checking of the environment that a user belongs to. Second, which is the interface used by the user? The checking of interface is to make sure that the database will remain secure after being accessed by different users, or in other words the interface is spam/virus free platform that is being used to access the database. Third and the most important function of this model is the dynamic (incremental or decremental) check of roles and assigning of permissions. The model checks the access pattern of the user within every n minutes interval and decides whether the role and permissions assigned to the user should be incremented or decremented. If the user is active and using the database without any harm or threat, the system will increase its access permissions to allow the user to access more relevant data. And if the user is found not very secure for the database in terms of its usage, the permissions will be decreased or even the user may get a notification to leave the system. Or even the system will directly throw out the user considering the fact that the nominated user is causing some harm or threat to database. There should be a continuous check on the access pattern of the user to increase the safety of database.

7. CONCLUSION

In this paper we 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. However practical access control mechanism is complex to be enforced in every database environment. Access control can not be sufficient enough for the security of the database, but the combination of EBDACM with normal access control can make it more reliable and secure enough to ensure the privacy protection. 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.

8. REFERENCES

[1] E.Bertino, S.Jajodia, P.Samarati. "Supporting Multiple Access Control Policies in Databases System". In: IEEE Symposium on Security and Privacy. Oakland, California. 1996.

[2] X. LIU, Z. HAN et al., "The Extended Confidential Level of BLP Model" Proceedings of KES 2005, Knowledge-Based Intelligent Information and Engineering Systems PartIV, Pps257-262

[3] H. Qilong, H. Zhongxiao "Real-time Optimistic Concurrency Control based on Transaction Finish Degree". Journal of computer science. 2005,1(4):471~476

[4] K. Ramamritham, S.H. Son, L.C. DiPippo, "Real-time databases and data services, Real-Time Systems" Journal 28 (December) (2004) 179–215.

[5] C.A. Ardagna, M. Cremonini, E. Damiani, S.D.C. di Vimercati, P. Samarati, "Supporting location-based conditions in access control policies". In ASIACCS, Taipei, Taiwan, 2006, pp. 212–222.

[6] O. Technology, Oracle multimedia, <http://www.oracle.com/technology/ products/intermedia/index.html>, 2007.

[7] Alliance, Image Web Service, <http://aspalliance.com/404_image_web_ service>, 2008.

[8] http://en.wikipedia.org/wiki/Role-based_access_control

[9] R.S.Sandhu, E.J.Coyne, H.L.Feinstein, C.E.Youman, "Role-Based Access Control Models", Computer, vol 29, pp. 38-47, 1996.

[10] R. Agarwal, J. Kiernan, R. Seikant, Y. Xu. "Hippocratic databases". Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002, pages 143-154.

[11] P. Huanchun, G. Jun, Y. Xiaojun. "Dynamic Purpose-based Access Control". International Symposium on parallel and Distributed Processing with Applications 2008. IEEE Computer Society, Pages 695-700.

[12] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman. "Role-based Access Control Models". IEEE Computer, volume 29 issue 2, pages 38-47, 1996.

[13] Q. Ni, A. Trombetta, E. Bertino, J. Lobo. "Privacy-aware Role based Access Control". Symposium on Access Control Model and Technology, ACM, pages 41-50, 2007.

[14] Z. Lingli, L. Shuai, L. Junsheng, X. Haicheng. "A Dynamic Access Control model Based on Trust". 2nd IEEE Conference on Environmental Science and Information Application Technology, ESIAT 2010, pages 548-551.

[15] J. W. Byun, N. Li. "Purpose based Access Control for Privacy Protection in relational database systems". The International Journal on Very Large Data Bases, volume 17 issue 4, pages 603-619, 2008.

[16] Z. Hui, f. Zhiyi, S. Lijun, Z. Dan. "Domain-based Access Control for Collaborative E-Commerce System". 2nd IEEE International Conference on Pervasive Computing and Applications, 2007. ICPCA 2007.

[17] Z. Zhou, X. Renzuo. "A Context-Aware Access Control Model for Pervasive Computing in Enterprise Environments". 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08.

[18] H. Hua, L. Ande. "Actor and Trust-based dynamic access control model in universal Computing Environment".IEEE Second International Symposium on Intelligent Information Technology Application, 2008. IITA '08.

[19] C. Pang, G. Dong, and K. Ramamohanarao. Incremental maintenance of shortest distance and transitive closure in ¯rst-order logic and sql. ACM Trans. Database Syst., 30(3):698{721, 2005.

[20] C. Pang, X. Zhang, Y. Zhang, and K. Ramamohanarao. Maintenance of access roles in sql. In Technical Report, 2005.

[21] P. Chaoyi, H. David, M. Anthony. "Managing RBAC States with Transitive Relations" ASIACCS'07, March 20–22, 2007 ACM

[22] R. Indrakshi, K. Mahendra, "Towards a location-based mandatory access control model" computers & security Science Direct 25 (2006) 36 – 44

[23] M. J. Moyer and M. Ahamad, "Generalized role based access control," in Proceed-ings of the 2001 International Conference on Distributed Computing Systems, Mesa, AZ, Apr. 2001

[24] S.-H. Park, Y.-J. Han, and T.-M. Chung, "Context-Role Based Access Control for Context-Aware Application," HPCC 2006, LNCS 4208, pp. 572-580, 2006.

[25] X. Huang, H. Wang, Z. Chen, and J. Lin, "A Context, Rule and Role- Based Access Control Model In Enterprise Pervasive Computing Environment," in 1st International Symposium on Pervasive Computing and Applications, Aug. 2006, pp. 497-502.

[26] 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 /11