Role Based Access Control Model for Law Information System

Ekong, U.O. & Sambo, Emem.
Department of Computer Science
Faculty of Science
University of Uyo
Uyo, Akwa Ibom State, Nigeria.
uyinomenekong@uniuyo.edu.ng, emem4mbo@yahoo.com.

ABSTRACT

In today's world, people depend on computers to create, store, and manage critical information. Information protection has become increasingly important as a result of the rapid development and widespread deployment of computers and smart mobile devices in disseminating information. Litigation on the other hand, often involves sensitive matters such as criminal prosecutions, bankruptcy petitions, malpractice suits, discrimination cases, and patent disputes. Legal information involves collection of confidential information that should be shielded from public view to protect the safety of witnesses, the privacy of litigants, and the integrity of the adjudicatory process. With the increase in the large amount of legal data to be processed and stored coupled with the challenge of safeguarding these legal data from unauthorized access in the database has given rise to the need to protect these information. To overcome this problem a well-known access control model known as Role Based Access Control (RBAC) models is employed. This model has generated a great interest in the security community as a powerful approach to security management and ability to reduce administrative expenses. In this paper a RBAC model is developed with some special features integrated in the model such as encryption technique and controlled access time to improve on the existing system.

Keywords: RBAC, Encryption, Security, LIS, Database.

CISDI Journal Reference Format
Ekong, U.O. & Sambo, Emem. (2017): Role Based Access Control Model for Law Information System. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 8 No 2. Pp 27-34.
Available online at www.cisdijournal.net

1. INTRODUCTION

In today's world, people depend on computers to create, store, and manage critical information. Thus, it is important that computers and the information they store are accessible and available when needed. It also is important for users to protect their computers and information from loss, damage, and misuse. Sensitive data and information such as credit records, employee and user information, and purchase information should be well secured. A computer security risk has to do with any event or action which can cause a loss of or damage to computer hardware, software, data, information or processing capability. An intentional breach of computer security often involves a deliberate act that is against the law. Any illegal act involving a computer generally is referred to as a computer crime. The term cybercrime is an online or Internet-based illegal act.

Information protection in multi-user computer systems has become increasingly important as a result of the rapid development and widespread deployment of computer systems in our daily life. Prevention, detection and recovery are the most common protection measures used in computer system today. The prevention measure is regarded as the traditional core of computer security, which usually attempts to achieve three security goals: confidentiality, integrity, and availability. These security goals are concerned with prevention of unauthorized disclosure of information, unauthorized modification of information, and unauthorized withholding of information, respectively.

Role Based Access Control is defined as $M \subseteq Users\ X\ Permissions$ (1)

Where
Permission represents an authorization to access and perform an operation on an object while a user is a human user that wants to access the system. A user can only have access if he has a right permission

$U \in User\ has\ P \in Permissios: (U,P) \in AC$ (2) (Boadu and Armah, 2014).

Access control mechanisms are used to implement access control policies, and ensure that users' requests to access resources are only granted if those requests are authorized by the policies. It validates the rights of users against the set of the authorization rules and states to perform the operation on the database, (Hingwe and Bharu, 2016). RBAC is based on three major rules which include; role assignment, role authorization, and role permission (Hingwe and Bhanu, 2016).

On the other hand, Legal Information System is defined as a computer system that is designed to manage the legal system's administrative and legal information effectively. It controls and manages all the activities that relate to information processing, case filing, court records, dockets and calendar for court hearings, (Kamran, 2015). With increase in the large amount of legal data to be stored, and the challenge of developing RBAC model to handle complex security needs in legal Information System, along with protecting the privacy and sensitivity of the data, cryptographic technique can be employed to RBAC model to encrypt the data in such a way that only users who are allowed to access the data as specified by the access control policies are able to do so. The authorized users who satisfy the access policies will be able to decrypt the data using their private key and none else will be able to reveal the data content without the key. With the access time, the legal information system accountability is sure.

The aim of this paper therefore, is to develop an extended Role Based Access Control model that will be used to prevent unauthorized accesses to legal information thereby providing additional security to the data in the database. In this paper, an encryption and access time module is included to the existing RBAC model to provide the additional security to information access.

2. LITERATURE REVIEW

The concept of RBAC has been used with a multi-user computer system and multi-application online system since the late 1960s and early 1970s. However, RBAC has rapidly emerged in the 1990s as a promising technology for managing and enforcing security in large-scale enterprise-wide systems, largely because of the non-existing enhancement in the traditional Mandatory Access Control (MAC) and Discretionary Access Control (DAC) used in many computer systems and networks, (Ferraiolo and Kuhu, 1992). Thus, RBAC is an alternative to traditional MAC and DAC policies that is currently attracting increasing attention, particularly for commercial applications.

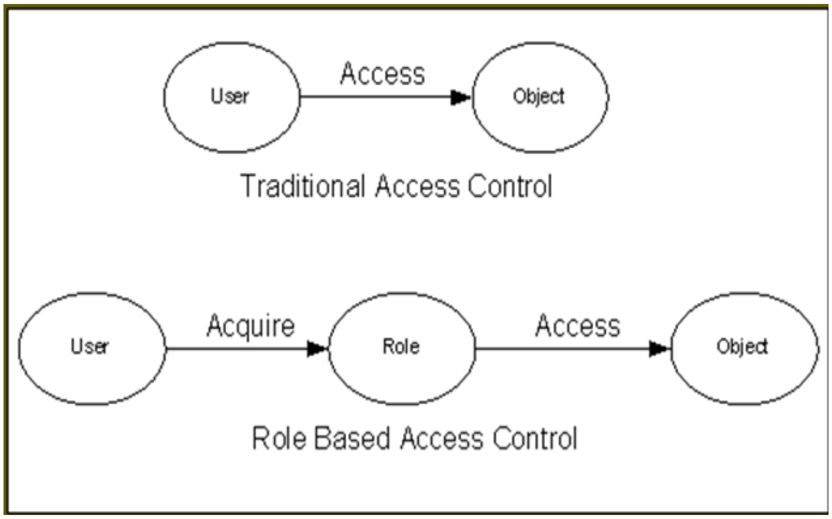

Figure1. RBAC versus Traditional Access Model (Klasky et al., 2013)

2.1 RBAC Models

RBAC is a family of reference models in which permissions are associated with roles, and users are assigned to appropriate roles. This greatly simplifies management of permissions. Role-based access control (RBAC) has served foundationally for newer models that enforce system security at various levels of improvement (Kibwage, 2015). The RBAC security model is abstract and general which is indicated by the many interpretations of the RBAC model provided by researchers of the existing interpretations, (Sandhu et al 1995). The basic idea of role-based access control is to include another level of indirection between the user to permission (or privileges) mapping. Roles thus break this mapping into two, the first part maps users to roles, while the second maps roles to privileges, (Andras, 2004).

Sandhu et al (1996) proposed RBAC model in 1996 which is made up of four models as shown in figure 2. There are RBAC0, the base model, is at the bottom, indicating that it is the minimum requirement for any system that professes to support RBAC. RBAC1 and RBAC2 both include RBAC0, but add independent features to it. They are called advanced models. RBAC1 adds the concept of role hierarchies (situations where roles can inherit permissions from other roles). RBAC2 adds constraints (which impose restrictions on acceptable; configurations of the different components of RBAC). RBAC1 and RBAC2 are incomparable to one another. The consolidated model, RBAC3, includes RBAC1 and RBAC2 and, by transitivity, RBAC0.

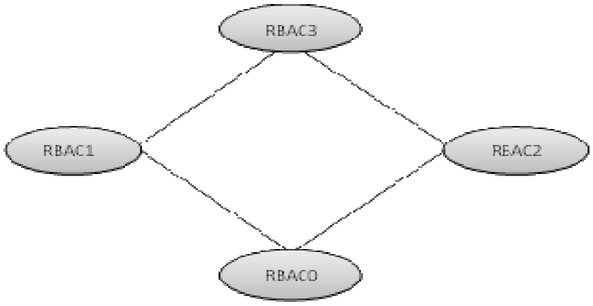

Figure2. The Family of RBAC Model (Chen, 2011)

The RBAC0 could be the simplest base model and it contains core concepts regarding the RBAC architecture, (Sunitha and Basu, 2014). The base model RBAC0 consists of three sets of entities called users (U), roles (R), and permissions (P). A user in this model is a human being. The concept of a user can be generalized to include intelligent autonomous agents such as robots, immobile computers, or even networks of computers. For simplicity, we focus on a user as a human being. A role is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. Permission is an approval of a particular mode of access to one or more objects in the system. The terms authorization, access right and privilege are also known as permission. Permissions are always positive and confer the ability to the holder of the permission to perform some action(s) in the system. The nature of permission depends greatly on the implementation details of a system and the kind of system that it is, (Sandhu et al., 1996). RBAC0 extends the support for least privilege sessions are introduced. They correspond to a particular occasion, when a user signs on the system to carry out some activity. In this way, sessions add a layer of indirection between users and roles, so that users activate roles within the frame of a session. Users can have many sessions, and in each session they can have different sets of roles active. Whenever a session is terminated the roles activated for that session are revoked.

Role Hierarchies

RBAC1 extends the basic model with role hierarchies. Role hierarchies were thought to be a natural way to describe role relations reflecting organizational structure. The permissions assigned to a junior role are inherited transitively by the more powerful senior roles. Role hierarchies define an inheritance relation among roles. Inheritance has been described in terms of permissions; i.e., r1 “inherits” role r2 if all privileges of r2 are also privileges of r1. (Chen 2011).

Constraints

Model RBAC2 introduces the concept of constraint which is an important aspect of RBAC and are sometimes the principal motivation for RBAC. The RBAC2 model extends RBAC0 by adding constraints, which help to specify preconditions to role entry. With respect to RBAC0 constraints can apply to the UA and PA relations and the user and roles functions for various sessions. (Sandhu et al 1996). The constraints added are also known as separation of duty which is divided into two parts, static separation of duty and dynamic separation of duty. Static Separation of Duty (SSD) that defines mutually disjoint user assignments with respect to sets of roles (Andras, 2004) while dynamic separation of duty limits the availability of the permissions over a user’s permission space by placing constraints on the roles that can be activated within or across a user’s sessions, (Kuhn, 2003).

Consolidated Model

RBAC3 uses RBAC1 and RBAC2 to provide both role hierarchies and constraints. There are several issues that arise by bringing these two concepts together. Constraints can apply to the role hierarchy itself, the role hierarchy is required to be a partial order. This constraint is intrinsic to the model. Two or more roles can also be constrained to have no common senior (or junior) role. These kinds of constraints are useful in situations where the authority to change the role hierarchy has been decentralized, but the chief security officer desires to restrict the overall manner in which such changes can be made.

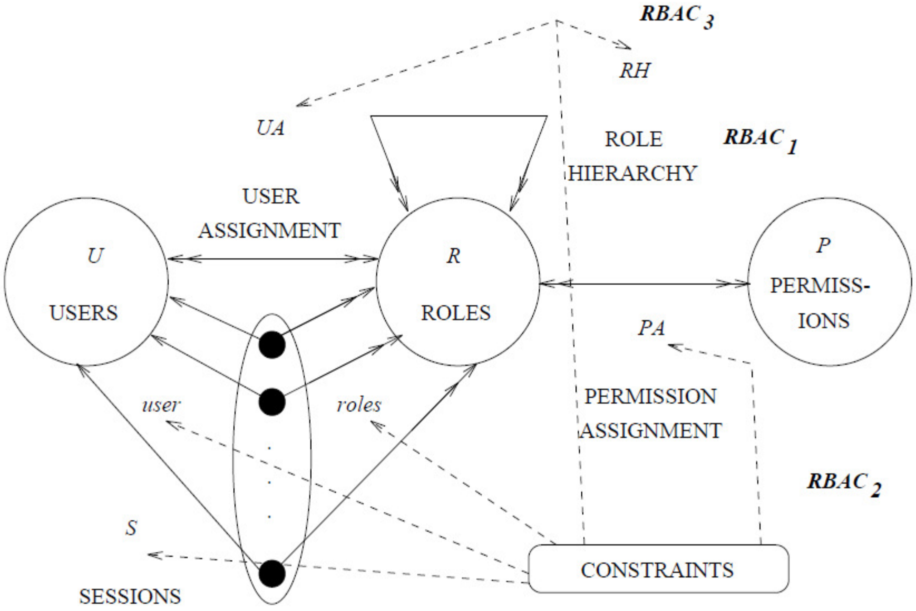

Figure3. The family of RBAC models (Sandhu et al 1996)

Using roles to determine and manage access permissions allows system administrators to better incorporate least privilege and separation of duties into administrative policies. As discussed, RBAC exists in many forms, but even its simplest form is an improvement over alternative methods. “RBAC features such as policy neutrality, principle of least privilege, and ease of management makes RBAC models a suitable candidate. Such models can express both DAC and MAC policies, as well as user-specific policies. (Gallaher et. al., 2002).

2.2 Some Extended Role Based Access Control

Role Base Access Control Administrative

To manage roles, users and their inter-relation in some large enterprise wide systems, where there is number of roles, users and information objects maybe formidable tasks for security administrators. Administrative RBAC is an alternative solution to the problem. ARBAC includes building and updating if RBAC information such as finding the roles and constructing role hierarchy, User Role Assignment (URA) and Permission Role Assignment (PRA), (Sejong and Seog, 2001). The drawbacks of RBAC are relaxation of complexity of administration unable to solve the fundamental problems of administration.

Task Role Based Access Control Model

T-RBAC is an improved RBAC model that solves the problems of general RBAC Model such as role hierarchy. The fundamental problem of RBAC administration, including building and managing RBAC schema information, is lack of information of interrelation between change of real world and change of RABC schema information. Security administrator needs an efficient user interface that supports real world style access control. To solve the problem, an improved RBAC model, task-role-based access control (T-RBAC) model is adopted. Permissions are assigned to related tasks, and tasks are assigned to related roles in T-RABC. Task-role assignment deals with the permissions between tasks and roles (Wang and Jiang, 2015).

T-RBAC is an integrated model of role-based access control and activity-based access control models based on task classification. There are 4 classes that have different access control characteristics in the companies. If a user U1 has tasks that belong to class S, their related access rights are inherited to user who has a higher job position than U1 in the organization structure. Tasks that belong to class W which is related to workflow and show the characteristics of an active access control model. Tasks that belong to class P are private ones; they do not have inheritance characteristic and related with workflow. Class A has characteristics of class S and class P. Class W and class P do not have inheritance characteristics.

The major difference between T-RBAC and RBAC is that the access rights are assigned to task in T-RBAC, but access rights are assigned to role in RBAC. In the real world access rights are needed for the user to perform tasks. So assigning access rights to task is reasonable. Another difference is the role hierarchy. The supervision role hierarchy (S-RH) is employed instead of general role hierarchy. In the S-RH, higher role does not inherit all access rights of the lower role in the role hierarchy. Only access rights of class S and class A are inherited from lower role to the higher role.

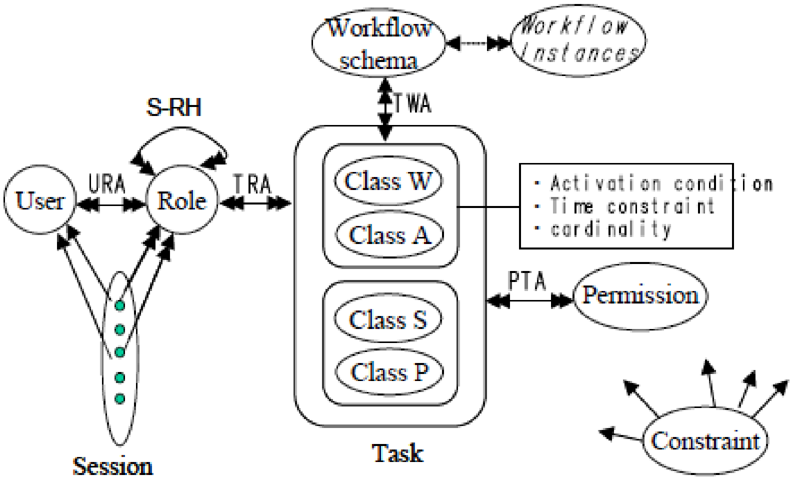

Figure4. T-RBAC Model, (Sejong and Seog, 2001)

Dynamic Role Based Access Control

Dynamic RBAC was implemented to solve the problem of permissions which required to be executed by sequential order in RBAC, (Junzheng et. al., 2011) described the traditional as follows; if one subject accesses one object, the subject has the operational authority which has access operation. Users have certain authority by role and corresponding operation without Implementing operation environment thus easily bringing security hidden. On the base of the original RBAC model, the dynamic RBAC model increases a module of dynamic constraint, which makes the dynamic constraints executed by sequential order. This point is not reflected in the general RBAC, when the role subordinate enters the system, he has the permission of remittance where he can execute permission which can result in security risk. To avoid the occurrence of this security problem concept of dynamic constraint is introduced. Dynamic RBAC examines the current permission which will be executed on the flow, finding the result that the permission can be executed or not. The permissions which do not need to be executed on the sequential order is the static permission.

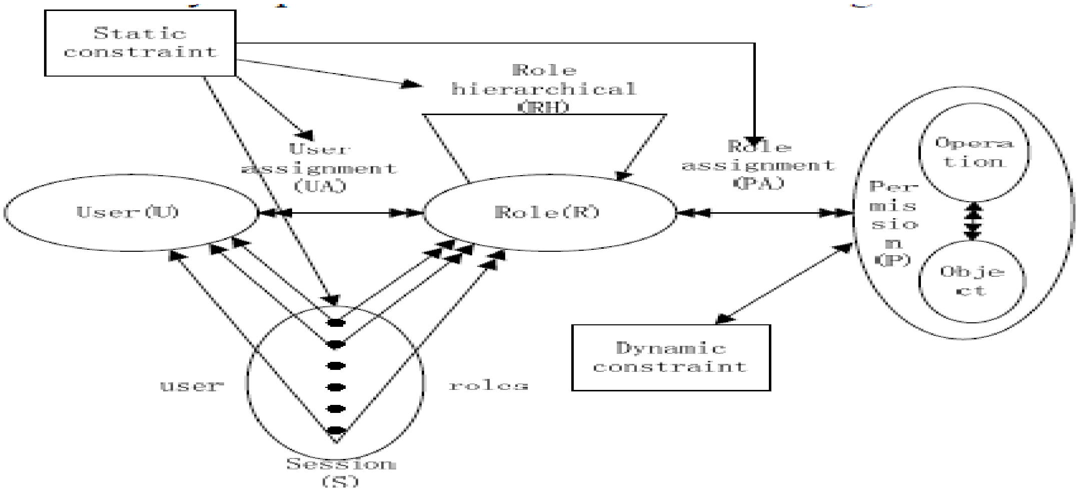

Figure5. DRBAC model (Junzheng et. al., 2011)

3. THE PROPOSED RBAC MODEL

The framework of the proposed RBAC-LIS model is with the integration of cryptographic technique and access time is shown in figure 6. These features will be added to the Dynamic RBAC Model proposed by (Junzheng et. al., 2011), in figure 5 to give a stronger constraint because of the sensitivity of legal information. RBAC-LIS defines a set of users U, a set of roles R, a set of permissions P, a user-role assignment relation UA ⊆ U×R and a permission-role assignment relation PA ⊆ P×R which refers to such sets and relations as components of RBAC. Roles(u) for the set of roles to which a user u is explicitly assigned by the UA relation; that is, Roles(u) = {r ∈ R : (u, r) ∈ UA}. Similarly, Roles(p) for the set of roles to which a permission p is explicitly assigned by the PA relation; that is, Roles(p) = {r ∈ R : (p, r) ∈ PA}. Given r ∈ R, Prms(r) to denote the set of permissions for which r is explicitly assigned, and for R0 ⊆ R, Prms(R0) to denote the set of permissions for which the roles in R0 are explicitly assigned. That is, Prms(r) = {p ∈ P: (p, r) ∈ PA} and Prms(R0) [r ∈ R0 Prms(r).

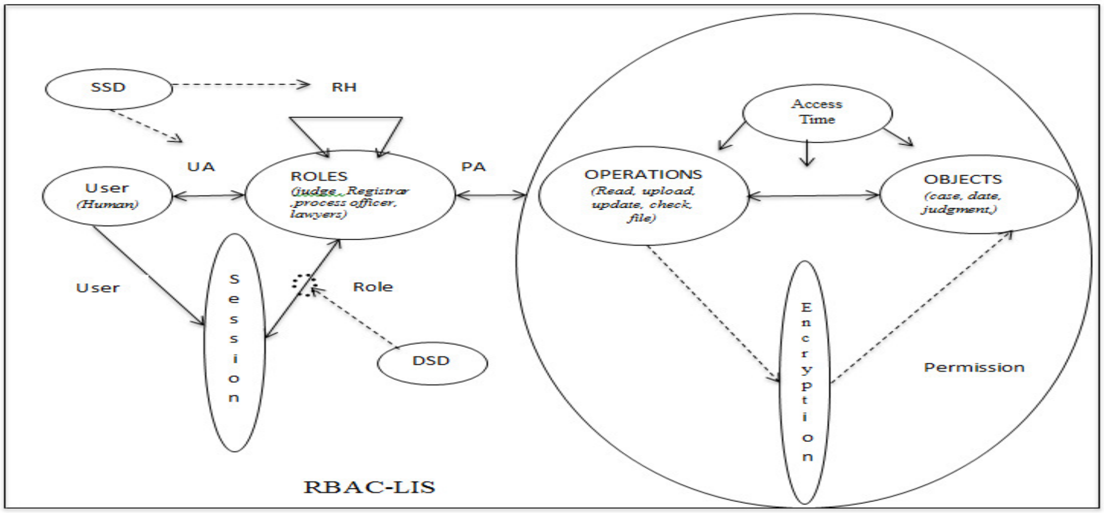

Figure 6. Proposed Framework for RBAC-LIS

A user is defined as a human being that wants to access the system. In order for this user to access the system, he or she must be assigned to a role. Role is the job function to the legal system given by an assigned user with some authority and responsibility for the given task such as the Role of a Judge, Registrar, process officer, clerk and the lawyers. The Role comes in between the user that wants to access the object. No user can access this system without being assigned a role. As an example, both a judge and a judge clerk may be authorized in the legal information system, but under no circumstances should they be authorized with the same permissions. While a clerk might be able to access legislative decisions and documents in order to perform paper-work, he/she should not be able to access evidences for a case. Even among judicial actors who belong to the same level of hierarchy, authorization procedures are essential. While a judge should be able to access information regarding a case assigned to her, she should not be able for a case which is assigned to another judge.

User Assignment (UA) relations the arrows indicate a many-to-many relationship that is a user can be assigned to one or more roles, and a role can be assigned to one or more users)

Permission Assignment (PA) relations is a many-to-many relationship between roles and permissions which means that a role can have many permissions and permission can be assigned to many roles.

Session is established when user activates the role they belong to. When a user login to the system to carry out some activities, sessions are activated between users and roles, so that users activate roles within the frame of a session. Users can have many sessions, and in each session they can have different sets of roles active. Whenever a session is terminated the roles activated for that session are revoked.

Role Hierarchy is the inheritance relation among roles such that role1 inherits role2 and the permission of role2 belongs to role1.

Static Separation of Duty (SSD) is to enforce constraints on the UA such that if a user is assigned to one, the user is prohibited from being a member of another role. That is to say no user can be simultaneously assigned to both roles in SSD.

Dynamic Separation of Duty (DSD) enforces constraint on the role that can be activated within a user's session. That is to say a user is not allowed to resume both roles at the same time.

Permission is an approval to perform an operation on one or more protected objects. An **operation** is an executable image of the system upon which a user executes some function. Such as update, read, upload, and file. An object is an entity that contains the legal information, such as the cases, legal records, judgment records.

Encryption technique for this method an encryption technique will be used, in such a way that only the users who are allowed to access the data specified by the access policies will be able to do so. The authorized users can access the data using their encryption key and no one can decrypt or reveal the information without the key. The authorized user with assigned permission allocated by its role cannot have access to some objects that are encrypted. Permission will be granted to users who qualify the role and permissions can also be revoked from existing users of the role, so when a user is a role or roles the key will be generated to the user and when a user is revoked from the role, the key will also be revoked alongside with the user.

Access Time is the time between the start of an access attempt to the object and the successful access granted to the user. It will be used as a form of an identifier which will be used to track a user or an intruder who has modified the information in the database. In a situation whereby different users assigned to a role may have access to a permission allocated by that role and it is discovered that some information within the permission of that role is modified, the access time will identify all the users that had permission to the modified object and the time the object was modified, and who exactly access the object at that time.

4. CONCLUSION

RBAC model is a highly desirable goal for addressing the key security requirements of cloud-based applications in general. Roles can be assigned to workflow tasks so that a user with any of the roles related to a task may be authorized to execute it. However, the challenge is to develop an improved RBAC model to handle the complex security needs of the legal Information System which involves encryption and access time. To protect the privacy of the data, cryptographic technique can be employ to encrypt the data in such a way that only users who are allowed to access the data as specified by the access control policies will be able to do so. The authorized users who satisfy the access policies will be able to decrypt the data using their private key. And none else will be able to reveal the data content without the key. With the access time the system accountability is sure.

REFRENCES

1. Abhishek M. Suyel N. & Samir N. (2014). *Taxonomy and Classification of Access Control Models for Cloud Environments,* Z. Mahmood (ed.), Continued Rise of the Cloud, Computer Communication and Networking (part 1) Doi: 10.10071978-1-4471-6452-4-2, ISBN: 978-1-4471-6452-4, pg 23-53, Springer, Verlag Lodon http://www.springer.com/978-1-4471-6451-7
2. Andras B. March (2004). *Role-based access control policy administration,* Technical Report, University of Cambridge, ISSN 1476-2986 http://www.cl.cam.ac.uk/ .
3. Boadu E.O. and Armah G.K. (2014). Role Based Access Control (RBAC) Based in Hospital Management, International Refereed Journal of Engineering and Science (IRJES), Vol.3, Issue 9, ISSN 2319-183X, pg 53-67.
4. Chen L. (2011). *Analyzing and Developing Role-Based Access Control Models*, Ph.D thesis, University of London Information Security Group, Royal Holloway, https://repository.royalholloway.ac.uk/file/817519d1-0731-c09f-1522-e36433db3d2c/1/liangcheng.pdf
5. Ferraiolo, D.F. and Kuhn R. (1992). *Role Based Access Controls*. In the 15th NIST-NCSC National Computer Security Conference, Baltimore, USA, Pg. 554-563.
6. Gallaher M. P, Connor A. C and Kropp B. (2002). *The Economic Impact of Role-Based Access Control*. National Institute of Standards and Technology Acquisition and Assistance Division, Gaithersburg, USA. MD 20899-0001.
7. Hingwe K. and Bhanu S. M. (2016), *Hierarchical Role-Based Access Control with Homomorphic Encryption for Database as a Service.* In: Satapathy S., Joshi A., Modi N., Pathak N. (eds) Proceedings of International Conference on ICT for Sustainable Development. Advances in Intelligent Systems and Computing, vol. 409. Springer, Singapore DOI: 10.1007/978-981-10-0135-2 43, e-ISBN: 978-981-10-01135-2, Pg. 437-448 http://arxiv.org/pdf/1603.00572.pdf.
9. Zheng J., Zhang Q., Zheng S. and Tan Y. (2011). *Dynamic Role-Based Access Control Model.* Journal of Software, Vol. 6 No. 6, DOI: 10-1145/2523649.2523676, Pg. 1096-1102.
10. Kamran K. (2015). *Legal Information System: A Model Framework for Indian High Courts.* Journal of Social Sciences and Humanities, vol 1. Pg. 38-47, DOI: 10.1234.67/sshj.1003.
11. Kibwage S. (2015). *Role-Based Access Control Administration of Security Policies and Policy Conflict Resolution in Distributed Systems*, Ph.D Dissertation, Nova Southeastern University. http://nsuworks.nova.edu.
12. Klasky H. B., Williams P.T., Tadinada S.K. and Bass B.R. (2013), *A Role-Based Access Control Schema for REAP.* Computational Sciences and Engineering Division, ORNL, USA.
13. Kuhn R. (2003), *Role Based Access Control.* American National Standard for Information Technology, American National Standards Institute, Inc. http://csrc.nist.gov/rbac/rbac-std-ncits.pdf
14. Sandhu R.S., Coynek E.J., Feinsteink H.L, and Youman C. E. (1996) *Role Based Access Control Models,*IEEE Computer, vol. 29, No.2, Pg. 38-47. http://csrc.nist.gov/rbac/sandhu96.pdf
15. Sejong O. and Seog P. (2001). *An Improved Administration Method on Role-Based Access Control in the Enterprise.* Journal of Information Sciences and Engineering, 17, 921,944.
16. Sunitha B.S, and Basu A. August (2014). *Review of Role Based Access Control Method for Securing User Space in Cloud Computing*, International Journal of Computer Trends and Technology (IJCTT), Vol. 14, No. 1, Pg. 22-25, ISSN 2231-5381, http://www.ijcttjournal.org.
17. Wang P. and Jiang K. *(*2015), *Task-role-based Access Control Model in Smart Health-care System* MATEC Web of Conferences. http://www.matec-conferences.org.