I.J. Computer Network and Information Security, **2014, 3, 1-9**
Published Online February 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2014.03.01

Enhanced Role Based Access Control Mechanism for Electronic Examination System

Adebukola Onashoga, Adebayo Abayomi-Alli, Timileyin Ogunseye
Department of Computer Science, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria
Email: onashogasa@unaab.edu.ng, abayomiallia@unaab.edu.ng, ogunseye.timileyin@gmail.com

Abstract — Over the years, e-learning and e-examination has become standard in many institutions of higher learning. It has been observed that examination questions and results can be easily intercepted by invalid users, thus the security of resources shared among valid users is not guaranteed. In order to solve these problems as it relates to access control, a Role based Examination System (RBES) was designed, developed and evaluated. RBES attempted to solve the security issue by the combination of two authentication techniques: text-based authentication and graphical password authentication. The Text-based authentication utilizes two text-based parameters namely the username and password. The graphical password authentication makes use of a finite set of controls (RBES chooses radio buttons) which are identified by numbers. These numbers constitute the password used for graphical authentication. To improve on resource sharing among users in the examination system, RBES proposes role management (role creation, role update, role removal) and user management (user creation, user update and user removal). The developed system made use of asp.net, C#, IIS server, WAMP server, Mysql and other tools for its development. RBES was tested by some legitimate and illegitimate users and the performance of the system was found to be satisfactory, hence RBES shows an efficient and reliable scheme that can be deployed in any examination or e-learning system. Finally the potential threats to the system were modeled and the use of weak passwords was found to be the most likely threat the system could be vulnerable to.

Index Terms — Access-control, Authentication, Authorization, RBES, Role, Privileges, Examination.

I. INTRODUCTION

The security of data and information in any information system can be used as a measure of determining the performance of the system. The confidentiality, integrity and authenticity of information constitute the basic elements of security which is used as a standard of determining how secured the system is. There is need for a secured system of exchanging confidential and sensitive information in an information system [1].

The Examination system has evolved through the years, the manual or hand written examination gradually evolved to the use of optical mark reader (OMR) and optical character recognition (OCR). The optical mark reader was very fast at computing result but lacked integrity with attendant errors [2]. This system gradually evolved to the use of computer based examination generally known as the Electronic Examination system. The Electronic examination system was more effective and efficient than the Optical mark reader. It has the capacity of computing result immediately after submission. The Electronic examination was a much more improved system but lacked confidentiality [3]. Information (questions and results) can be intercepted during data transfer between different parties. Hence, the need to restrict access from invalid users beckoned.

Access control involves users providing the system with their valid identity and the system verifies the supplied identity in order to determine eligibility for access and the allowed activities of a legitimate user [4], [5]. Access control specifies the need for subject, target and rules as a means of enhancing security [6]. The security provided by access control can only prevent illegitimate users but cannot restrict the activities of legitimate users.

Role-based access control is a framework for controlling user access to resources based on the allotted roles. Each user is allotted a particular role and each role is assigned set of privileges to which the user can only get access. The Role Based access control technique could therefore provide the adequate resource management technique required for an electronic examination system. Hence, this paper proposes a secured electronic examination system, which plays a significant role in the provision of better examination system where information in form of questions and results can be processed without fear of data integrity or compromise. The proposed system is aimed at improving security in an electronic examination system and ensuring proper management and sharing of resources between the systems users.

The main benefit of using RBAC is an improvement in the manageability of the system. Because an administrators only have to maintain a list of who is entitled to exercise which role. Role Based Access Control (RBAC) was first conceived in the 1990’s. It is a mature and widely used model for controlling access to operating systems and software [7]. Although there is

Manuscript received October 23, 2013; revised November 18, 2013; accepted December 2, 2013.

no consensus to the definition of Role Based Access control (RBAC), RBAC is a framework for controlling user access to resources based on roles. It can significantly reduce the cost of access control policy administration and is increasingly widely used in large organizations [8]. RBAC allows access decisions to be based on the roles that individual users have as part of an organization. RBAC control method, exploit the association between users and permissions through the assignment of user's roles. Users can be assigned to roles, having associated permissions and thus, users acquire permissions by having roles [9], [10].

The central notion of RBAC is that permissions are associated with roles, and users are assigned to appropriate roles. This greatly simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as the need arise. Thus a role is properly viewed as a semantic construct around which access control policy is formulated. The particular collection of users and permissions brought together by a role is transitory. The role is more stable because an organization's activities or functions usually change less frequently. RBAC adds the notion of roles as a level of indirection between users and permissions. Roles are created based on job functions and/or qualifications of users. Permissions (i.e., privileges to access resources) are assigned to roles based on the requirements of job functions and/or the entitlement of qualifications. Users are made members of roles based on their job responsibilities and/or qualifications, thereby gaining permissions assigned to those roles. Role was defined as a semantic construct forming the basis of access control policy in [11]. It can represent specific task competency, such as that of a physician or a pharmacist or may also reflect specific duty assignments rotated through multiple users, for example, a duty physician or a shift manager. RBAC models and implementations conveniently accommodate all these manifestations of the role concept.

The remaining part of this paper contains the review of related works in section II, design methodology of the proposed role based examination system was presented in section III, section IV shows the design implementation for the system. The system was tested and the results were evaluated in section V while the paper was concluded with future research directions in section VI.

II. RELATED WORKS

Role based access control has been studied by different researchers and institutions. The various RBAC models, applications, problems and uniqueness have been analyzed by different researchers. Some of these researches and related works are discussed below.

One of the earliest proposals of RBAC is the Generalized Role-Based Access Control (GRBAC) model by [12]. It introduced the concept of environmental roles to capture contextual information from the underlying access environment.

GRBAC, however, requires the use of complex system architecture to support the extended roles [13].

Another notable proposal is the Temporal RBAC (TRBAC) model by [14] which extends the traditional RBAC model by introducing a temporal constraint into the access control specification to provide a mechanism to enforce time-dependent access control policies. Generalized Temporal RBAC (GTRBAC) model by [15], further extends the TRBAC model by introducing the notion of an activated role. More precisely, GTRBAC differentiates enabled roles, whose subjects can activate from active roles, which are being activated by at least one subject, for a more fine grained access control.

The Dynamic Role Based Access Control (DRBAC) model [16], specifically designed for access control adaptation in ubiquitous environments, which in turn, trigger an ‘event’ to cause a transition between the current role/permission set to a new role/permission set. However, as noted by the authors themselves, implementing DRBAC can significantly increase the complexity of the applications concerned. This is particularly troublesome for the resource-restricted devices typically seen in ubiquitous environments.

Context-aware access control model was proposed by [17], it considers location, time, and system resources as access control constraints. The role is activated only if all the constraints are satisfied. However, the model fails to consider the potential composite effects and the correlations between, these context attributes. In a similar approach, [18] divide the location information in a leveled manner. The work formalized the model and conducted a case study, but again, it only focuses on spatial context attribute.

The work of [19] tried to address the need for evaluating the effect of multiple contextual attribute on an authorization decision coherently. The model introduces the notion of risk-aware access control. The context information is used as the input to a risk assessment process to compute a risk value that is then fed into the authorization decision engine. However, the scope of the risk assessment is quite broad covering confidentiality, integrity and authentication, so the delay incurred in the risk value calculation may be quite large, which may adversely affect the performance of the underlying access control system. Whether this delay would decrease the system ability to promptly adapt its decisions to context changes is yet to be investigated. An access control framework that monitors patients after hospitalization was designed by [20]. The work enforces some security policies for a smart item application scenario. Authentication in the system is extremely maintained by third party. This system is one of the successful records of applications of access control in the health domain. A secured scheme for electronic result transfer using a hybrid XML security scheme was presented by [1]. The proposed scheme combines the XML signature and XML Encryption schema. The application basically converts the input data (result record) into XML document. The processed data is then converted into a mark sheet which is a presentable in form of students' result.

III. METHODOLOGY

This section introduces the proposed design of the Role-Based Examination System (RBES). An electronic examination system that uses a role based access control model. RBES adopts the concept of the role based access control framework in [11]. The proposed design is divided into two interrelated phases; the Authentication Phase and the Authorization Phase. The remaining part of this section describes each of these phases.

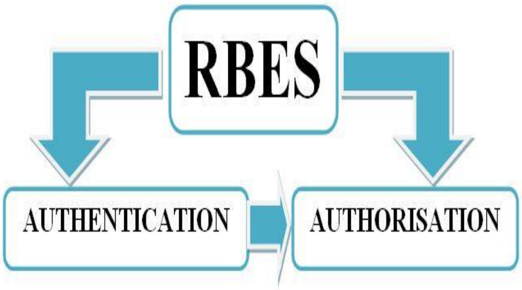


Figure 1. General Descriptions of RBES

A. *Authentication Phase*

Authentication is generally described as the way or means of permitting an object to a system resource. RBES uses two authentication techniques, namely: Text-based Authentication and Pattern / Graphical Authentication.

a. *Text-based Authentication*

This type of authentication uses two text-based user parameters to grant access to the system resources. Most software application and web application utilizes this technique for user authentication. In RBES, each user is identified with a unique username and a password to identify the user. RBES text based authentication follows the scheme in table 1. In order to enhance the security, RBES uses the MD5 hash algorithm to encrypt each user's data. The MD5 algorithm is a one way encryption or hashing algorithm that is used to prevent any form of data manipulation and insecurity from the backend. Data stored in the database can be viewed by the administrative officers, hence to prevent data manipulation and authorized access; the identity of each user (username and password) in RBES is encrypted before storing it in the database.

b. *Graphical Authentication*

This type of Authentication is often called pattern recognition passwords. Each user is identified with a specified pattern and this pattern is used to authenticate the user. RBES uses graphical authentication as an additional authentication measure. For this type of authentication, RBES provides nine (9) radio buttons which is uniquely identified with a number ranging from one (1) to nine (9) respectively.

TABLE 1. TEXT BASED AUTHENTICATION SCHEME.

```
User inputs username and password
// defining variables for authentication
Let RbesData represents Database object (describing database entities and attributes)
        and
Let RbesUser represents User object (describing user parameters).
If RbesData.password = RbesUser.password
    then grant resource access to user
  Else deny resource access to user
```

Table 2 shows the function for generating the graphical password. For example, RBES defines a user "admin" identified by a graphical password "129" which implies that user admin registered with radio button1, radio button 2 and radio button 9.

TABLE 2. FUNCTION GENERATE GRAPHICAL PASSWORD

```
1. Let count = 1;
2. Let password = " ";
3. If count <= 9  do 4 and 5;
4. If  button[count]  is selected then do 5
5. Password = password + button[count].value
     End.
```

The RBES function graphical authentication on Table 3 present the authentication policy that compares the password in the database (*RbesData*) and the extracted user password (*RbesUser*).

TABLE 3. GRAPHICAL AUTHENTICATION POLICY

```
User checks the radio Buttons
Let RbesData represents database object (describing
   database entities and attributes)
                 and
Let RbesUser represents User object (describing user
    parameters).
If     RbesData.graphicalpassword =
       RbesUser.graphicalpassword
     Then grant resource access to user.
Else deny resource to user
```

B. *Authorisation Phase*

Authorization is generally described as the means of granting operation on a particular privilege to a user or set of users identified by a role. RBES authorization phase is in two different tiers namely Examination tier and the Administrative tier. The sub units under the examination and administrative tiers are shown in Fig. 2.

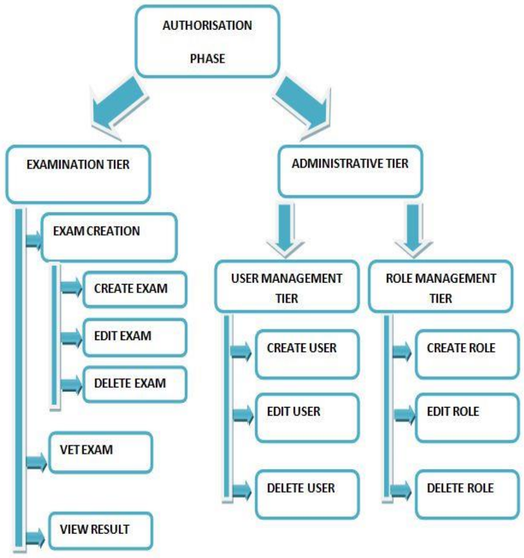


Figure 2. The Complete Authorization Model

C. *Examination tier*

This tier of RBES authorization deals with examination processing. It incorporates creation, manipulation and analyses of examination question and result. This tier has the following elements.

D. *Administrative Tier*

This tier manages all the administrative functions of RBES. Most of the Role-based Access control technique is described in this tier. This tier has the following elements. The Administrative tier makes use of sets of commands that are specifically for administrative purposes. They deal with user management, role management, addition and deletion operations. The administrative commands involve adding and deleting elements to a set of users or roles.

TABLE 4. ELEMENTS OF THE EXAMINATION TIER.

	Description	**Notation**	**Example**
U	Finite set of users	$U = \{u_1, u_2, u_3,\ldots, u_n\}$	“Timilehin”, “Ayoola”, “Adekoya”
C	Finite set of courses offered by the institution	$C = \{c_1, c_2, c_3,\ldots, c_n\}$	“csc421”, “csc234”, “phs102”
$U \subseteq \boldsymbol{U} * \boldsymbol{C}$	is the user-course assignment relation	$\{u, c\} \in \boldsymbol{UC}$	{“Adekoya”,”csc213”}, {“onashoga”,”csc321”}, {“shodiya”,”csc413”}

TABLE 5. ELEMENTS OF THE ADMINISTRATIVE TIER

	Description	**Notification**	**Example**
U	Finite set of users using the system	$U = \{u_1, u_2, u_3,., u_n\}$	Timilehin”, “Ayoola”, “Adekoya”
R	Finite set of roles available in the system.	$R = \{r_1, r_2, r_3,., r_n\}$	“lecturer”, “hod”, “Admin1”
P	Finite set of permissions available in the system.	$P = \{p_1, p_2, p_3,., p_n\}$	“create exam”, “vet exam”, “view result”
O	Finite set of objects accessible to a user available in the system.	$O = \{o_1, o_2, o_3,., o_n\}$	examination questions, examination result
S	Finite set of user session.	$S =\{ s_1, s_2, s_3,., s_n\}$	“create”, “edit”,” select”
OP	A finite set of operations available in the system.	$OP=\{op_1,op_2,op_3,., op_n\}$	
UR U* R	The user-role assignment relation.	$\{u, r\} \in \boldsymbol{UR}$	{“Adekoya”, “hod”}, {“Timilehin”, “Lecturer”}
PR P * R	The permission-role assignment relation.	$\{p, r\} \in \boldsymbol{PR}$	{“lecturer”, “create exam”},{“exams and records”, “view result”}

Adding an element to a set of users or roles uses the functions *Add User* and *Add Role,* which can be specified in a similar way; the same is true for adding an element to a relation (*Assign User* function adds to UR and *Grant Permission* function adds to PR, except that the latter uses nested tuple). The following functions/commands are used in RBES administrative.

Function Add User: This function is used to add a new user u to the system users U. It takes an argument u, checks the users table and ensures that the user u is not a member of users U and then inserts the user record into the users table.

Add User (user u):
Precondition: u∉ U
U = U + {u}

Function Add Role: This function is used to add a new role **r** to the system roles **R**. It takes an argument **r**, checks the roles table in order to ensure that the role **r** does not exist in the roles **R** table, and then inserts the new role into the roles database.

Add Role (role r):
Precondition: r∉ R
R = R + {r}

Function Assign Role to User: This function is used to assign role r to a particular user u. The function takes two arguments, the user u and the role r. The function checks the roles table and the users table in order to ensure that both arguments are available in the tables. The function also checks if the role r has not been assigned to the user u.

Assign User (user u, role r):
Precondition: ***u***∈ ***U*** , ***r***∈ ***R***
*[**u**, **r**]* ***UR***
UR = ***UR*** + *{[**u**, **r**]}*

Function Grant Permission: This function grants permission on an object **o** to a role **r**. The function takes three arguments consisting of operation **op**, object **o** and role **r**. The function checks the argument in their respective tables in order to ascertain their availability. Hence grants the role the specified operation on the permission.

Grant Permission (operation op, object o, role r):
Precondition: op OP, o O, r R
[[op, o], r] PR;
PR = PR + {[[op, o], r]}

Deleting an element is symmetric to adding an element, but possibly with two kinds of additional updates. First, if an element is deleted from a set, then from all relations defined using the set, all pairs that contain the deleted element must be deleted. Second, *Delete User*, *Delete Role*, and *Deassign* User may affect SESSIONS, because sessions are created by users and have active roles, and must satisfy the constraint that a session can have a role only if the user of the session is assigned that role. Specifically, *Delete User* may either delete associated sessions or leave the sessions to terminate normally; *Delete Role* and *Deassign* User have a third option of deleting only the specified role from the sessions. As in the standard, we formally specify only the first option, i.e., deleting all associated sessions, for all three operations, illustrating both kinds of additional updates; the other two options are simpler to specify.

Function Delete User: This function removes an existing user **u** from the users **U**. The function first iterates through the users table in order to check if the user **u** exist, if the user exists, the function calls another function Delete Session in order to remove the sessions already created by the user thereby removing both the user and also the user session. Then deletes the user from the users table

Delete User (user u):
Precondition: u U;
UR = UR - {[u, r]: r∈ R
For s Delete Session (u, s)
(//the function is defined next)
U =U- {u}

Function Delete Session: On creating a user, certain variables are assigned to the user to monitor and aid the user's activity in the system and also to guide against role conflict. Whenever a user is deleted from the system, the function ensures complete removal of the session data created for the user.

Delete Session (user u, Session s)
Precondition: s∈ S
S = S - {[u, s]}

Function Delete Role: This function removes an existing role **r** from the set of roles **R**. The function takes one argument which is the role **r** to be deleted. The function checks the role table in order to ensure that the role exists, and then removes the role from the roles table.

Delete Role (role r)
Precondition: r∈ R
PR = PR – {[op, o], R}: op OP, o∈ O
UR = UR – {[u, r]: u∈ U
R = R – {r}

Function Deassign User: This function de-assigns a role **r** from an existing user. The function performs de-assigning of role after ensuring that the role has been assigned to the user before.

Deassign User (user u, role r):
Precondition: u U, r R,
[u, r] UR;
For s in S
[s, u] SU, [s, r] SR:
Delete Session (u, s)
UR = UR - {[u, r]}

The Administrative tier is further divided into two sub-tiers which are the User Management Tier and Role Management tier.

E. User Management Tier

The user management tier performs all activities pertaining to RBES users. The user management tier has

3 major functions User Creation, User Modification and User Deletion.

1. Create User

The user creation deals with the creation of a new user. This tier follows the order below

a. Administrator specifies the new user details and the role to be assigned to the new user
b. Function *Add User* creates the new user
c. Function *Assign Role to User* assigns a role to the user
d. Function *Create Session* creates a Session for the user

2. User Modification

The user modification handles user role alteration. In a full academic system, users are subject to change in role. If the role of a particular user undergoes changing (e.g. from a lecturer to the head of department), the particular aspect alters the User Data and access rights through:

a. Administrator specifies the user to be altered and the new role granted to the user,
b. Function *Delete Session* clears user session,
c. Function *Delete User*, Deletes the previous user,
d. Function *Add User* create a new user using the specified data,
e. Function *Assign Role* to User assigns a new Role to the User,
f. Function *Create Session* creates a new session to the user.

3. User Deletion

The user deletion module handles deleting of a user from the Users. This module follows the steps below:

a. Administrator specifies the user to be deleted,
b. Function *Delete Session* deletes user session,
c. Function *Delete User* deletes the user specified.

F. Role Management Tier

The role and management tier handles Role Creation, Role Modification, and Role Deletion.

1. F c cf Role Creation

The role creation tier handles the creation of a new role. Each role created in the electronic examination system has to be provided with some default permission. In a standardized electronic examination system, all lecturers can create examination (i.e. supply questions for an examination, the session created ensure that each lecturer can only create exam in his or her own field. For example A mathematics teacher can only set mathematics question), All Head of department can vet the created exam in their department by Default but some can also create examination if they are also lecturers. This tier works in the following manner.

a. Administrator specifies the new role to be created and its default permissions
b. Function Add Role Creates the role
c. Function Grant Permission grants the role the permission.

2. Role – Permission Modification

Roles Created is also subject to change with time. If a created role has to be altered, in order to add or remove permission, it follows the following steps

a. The Administrator Specifies the role to be altered and the new set of permission to be granted
b. Function *Delete Role* deletes Previous Role
c. Function *Create Role* creates a new role
d. Function *Grant Permission* grants the role the new set of permission

3. Role Deletion

Roles deletion removes a role completely from the system. This aspect follows the order below

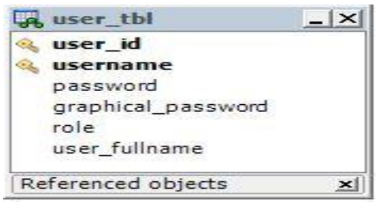
a. Administrator specifies role to be deleted
b. Function *Delete Role* deletes the role.

IV. IMPLEMENTATION

RBES was designed with C# on a Dot Net Framework 4.0 (the runtime environment for C#) with ASP.NET MVC, and WAMP server (My sql engine). The Internet information server (IIS) is installed as the webserver and for client side, the web browser is the only requirement.

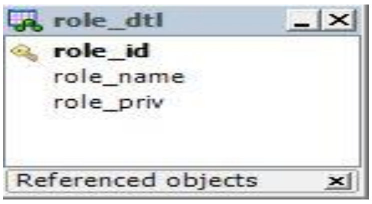
A. The Database Design

Fig. 3-5 shows the database design of RBES. Each table has a unique identifier generally known as a primary key which identifies each record in the table and the tables are well linked to each other hence creating a relationship between user, roles and privileges.



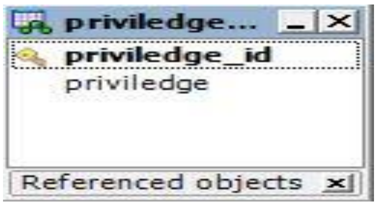
user_tbl
user_id
username
password
graphical_password
role
user_fullname
Referenced objects

Figure 3. User data in the database table.



role_dtl
role_id
role_name
role_priv
Referenced objects

Figure 4. Role Table



priviledge...
priviledge_id
priviledge
Referenced objects

Figure 5. Privilege Table

B. Choice of Programming Language

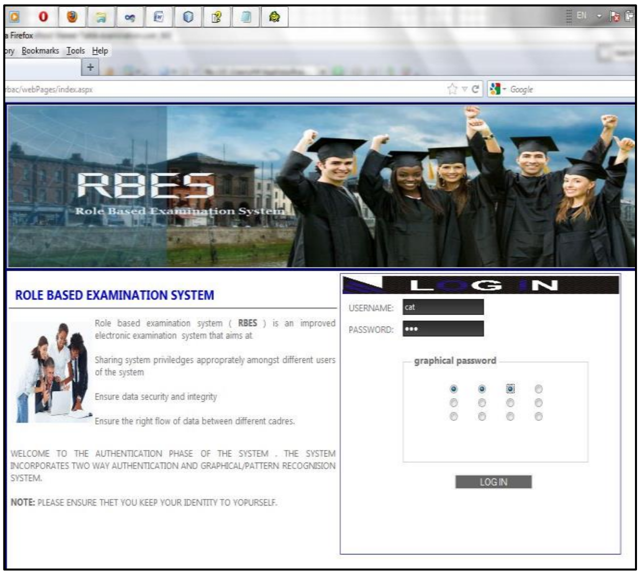
C# is used for the development of RBES due to the following reasons

- An object oriented programming language enabling encapsulation, polymorphism and inheritance;
- An easy to use programming language;
- It is widely used by many programmers, and it has the visual studio as an IDE which makes debugging easier and faster.

C. Graphical User Interface Design

This section contains snapshot of the developed system and information pertaining on each shot. Fig. 6 shows the first page of the application which is the Login page where authentication takes place. The login page has the following details:

- **Username:** The username of the user
- **Password:** The text-based password of the user which would be used to authenticate the user
- **Graphical Password:** An added authentication technique which will also be used as bases for authenticating the user. It contains 9 radio buttons which are identified by numbers; the user creates a pattern with this buttons which must tally with the database graphical password for the user.



ROLE BASED EXAMINATION SYSTEM

Role based examination system (**RBES**) is an improved electronic examination system that aims at

Sharing system priviledges approprately amongst different users of the system

Ensure data security and integrity

Ensure the right flow of data between different cadres.

WELCOME TO THE AUTHENTICATION PHASE OF THE SYSTEM . THE SYSTEM INCORPORATES TWO WAY AUTHENTICATION AND GRAPHICAL/PATTERN RECOGNISION SYSTEM.

NOTE: PLEASE ENSURE THET YOU KEEP YOUR IDENTITY TO YOPURSELF.

USERNAME: cat

PASSWORD: •••

graphical password

● ● ● ○
○ ○ ○ ○
○ ○ ○ ○

LOG IN

Figure 6. Authentication Page

The Username and password entered by the user is encrypted using MD5 algorithm before authenticating the user. The Username and password entered by the user is encrypted using Md5 algorithm before authenticating the user.

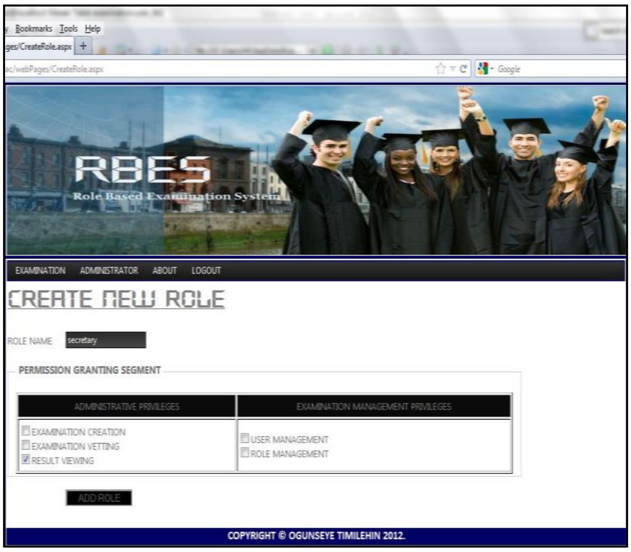
TABLE 6. ALGORITHM FOR USER AUTHENTICATION

```
Function AuthenticateUser( String Username , String
Password)
{     String user = Md5(username);
        String Pass = Md5(password);
             If( user = database.username and pass =
            database.pass)
       {   Grant Access;}
           Else
           { Deny Access;}
    }
```

RBES provides a means of adding roles to the system. The role creation page as shown in Fig. 7 provides the interface for entering the role attributes while the function *CreateRole* on Table 7 inserts the role record into the database**.**

TABLE 7. ALGORITHM FOR USER ROLE CREATION

```
      Function createRole (String rolename, String
                                            privileges
                                      {
      String Query = ("insert into role_tbl  values
      (Rolename, Privileges); ")
 }
```



CREATE NEW ROLE

ROLE NAME secretary

PERMISSION GRANTING SEGMENT

ADMINISTRATIVE PRIVILEGES	EXAMINATION MANAGEMENT PRIVILEGES
☐ EXAMINATION CREATION ☐ EXAMINATION VETTING ☑ RESULT VIEWING	☐ USER MANAGEMENT ☐ ROLE MANAGEMENT

ADD ROLE

COPYRIGHT © OGUNSEYE TIMILEHIN 2012.

Figure 7. Role creation page

V. SYSTEM TESTING AND RESULTS

RBES was tested with twenty (20) invalid users. Table 8 shows the result of the test conducted by different categories of invalid users.

A comprehensive threat modeling analysis was carried out which allowed RBES designers to systematically identify and rate the threats that are most likely to affect the system. The following were considered very important:

- Privacy of personal records,
- Confidentiality (protection from exposure),

- Integrity (protection from alteration) of exam questions and records.

Authentication and access rights to course materials, grade records, etc.

By using the using the STRIDE recommended by the Microsoft laboratory the following was identified as the most likely threats to RBES.

TABLE 9. LIST OF POTENTIAL THREATS TO RBES AND THEIR RISK RATES.

Threat	Risk Rate
1. Using weak password	0.97
2. Passing authentication credentials over unencrypted network links	0.93
3. Storing account credentials in clear texts	0.84
4. Using weak or custom encryption	0.75

VI. CONCLUSION AND FUTURE RESEARCH DIRECTION

A Role Based Access Control Examination system was designed, implemented and evaluated. The system exploits the structure of RBAC policies, constructing a system where each user has a role and each role has a privilege. This system provides a dynamic way of assigning resources to role and assigning roles to user without database manipulation and ensuring that each user can access the resources provided to a role. RBES was tested against common attacks and the performance was satisfactory but the use of attack trees in threat modeling is recommended in order to identify the specific attacks the system may be vulnerable to. Using Weak passwords was identified as the major vulnerability of the system to threat, hence strong passwords with a combination of alphanumeric and special characters are recommended. Future works will incorporate the use of biometric authentication for the system administrators and users in order to ensure continuous authentication in a cost effective way and avoid session hijacking.

TABLE 8. SYSTEM TESTING WITH INVALID USERS.

TEST	RESULT
Uses tried invalid text based password	RBES issued a message “invalid username/password”.
Users tried wrong graphical password	RBES issued a message “invalid username/password”.
Users tried to open URL of different RBES closed sessions.	RBES redirected back to Log in page
Users after been authenticated tried to view authorized pages	RBES redirected back to Log in page

REFERENCES

[1] Onashoga, S. A. and Sodiya, A. S. (2011). “A Confidential Electronic Result Transfer Using a Hybrid XML Security”. In Proc. of the 8th International Conference on Information Technology: New Generations (ITNG), Las Vagas, USA, August, 2011.

[2] Orchard, K. (1998). “The use of Optical mark reading (OMR) for census data collection”. 18th Population Census Conference, 26 – 29 August, 1998, East-West Center, Honolulu, Hawaii USA.

[3] Karami, M.; Heussen, N.; Schmitz-Rode, T. and Baumann, M. (2009). “Advantages and Disadvantages of Electronic Assessments in Biomedical Education”. World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany. 25(12): 61-64.

[4] American National Standards Institute (ANSI), “Role-based access control”, International Committee for Information Technology Standards (INCITS). ANSI INCITS Standard, 359 (2004), February 2004.

[5] Weil, T. (2012). “Role-Based Access Control”, INCITS CS1 Standards Series, Computer Science Colloquium, University of Denver, 13 January, 2012.

[6] Sandhu, R. (1996). “Issues in RBAC”. In Proc. of the ACM RBAC Workshop.MD: ACM Press, pp 21-24, 1996.

[7] Sandhu, R. S.; Cogne, E. J. and Feinstein, H. L. (1996). “Role-Based Access Control Models”. IEEE Computer, 29(1996):38-47.

[8] Sasturkar, A.; Yang, P.; Stoller, S. D. and Ramakrishnan. C.R. (2011). “Policy Analysis for Administrative Role Based Access Control”. Theoretical Computer Science, Elsevier, 412(44):6208-6234, October 2011.

[9] Schaad A., Moffett J., and Jacob J. (2001). The role-based access control system of a European bank: A case study and discussion. In Proceeding of 6th ACM Symposium on Access Control Models and Technologies (SACMAT).

[10] Schaad A. and Moffett J. D. (2002): A lightweight approach to specification and analysis of role-based access control extensions. In Proceeding of 7th ACM Symposium on Access Control Models and Technologies (SACMAT).

[11] Ferraiolo D. F., Sandhu R., Gavrila S., Kuhn D. R., and Chandramouli R. (2001): Proposed NIST standard for role-based access control. ACM Transactions on Information and Systems Security.

[12] Moyer, M. J. and Ahamad, M. (2001). “Generalized role-based access control”, In Proc. 21st International

Conference on Distributed Computing Systems (ICDCS '01), Washington DC, April 2001, IEEE Computer Society, pp. 391-398.

[13] Covington, M. J.; Fogla, P.; Zhan, Z. and Ahamad, M. (2002). “A Context-Aware Security Architecture for Emerging Applications”. In Proceedings of the Annual Computer Security Applications Conference, Las Vegas, NV, December 2002.

[14] Bertino, E.; Bonatti, P. A. and Ferrari, E. (2001). “A temporal role-based access control model”. ACM Trans. on Information System Security, 6(1):11-27, 2001.

[15] Joshi, J.; Bertino, E.; Ghafoor, A. (2002). “Hybrid role hierarchy for generalized temporal role based access control model”, in Proc. 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC '02), Washington DC., IEEE Computer Society 2002, pp. 951-956.

[16] Zhang, G. and Parashar, M. (2003). “Dynamic context-aware access control for grid applications”. In IEEE Computer Society Press, editor, 4th International Workshop on Grid Computing (Grid 2003), November 2003, Phoenix, AZ, USA. pp 101–108.

[17] Kim, Y-G.; Mon, C-J.; Jeong, D.; Lee, J-O.; Song, C-Y. and Baik, D-K. (2005). “Context-Aware Access Control Mechanism for Ubiquitous Applications”. Advances in Web Intelligence, Lecture Notes in Computer Science, 3528 (2005): 236-242.

[18] Chae, S.; Kim, W. and Kim, D. (2006). “Role-based access control model for ubiquitous computing environment”, Information Security Applications, Springer Berlin / Heidelberg, 3786 (2006): 354-363.

[19] Diep, N. N.; Lee, S-Y.; Lee, Y. K. and Lee, H. (2007). “Contextual Risk-Based Access Control”, Security and Management, 2007: 406-412.

[20] Evered, M. and Bögeholz, S. (2004). “A case study in access control requirements for a health information system”. In Proceedings to Australasian Information Security Workshop (AISW), Volume 32 of Conferences in Research and Practice in Information Technology.

Onashoga S. A. holds a Ph.D. degree in computer science. She is presently with the Department of Computer Science, Federal University of Agriculture Abeokuta, Nigeria where she teaches and supervises at undergraduate and post-graduate level. She is a chartered information technology practitioner and her main research interests include information security and data-mining.

Abayomi-Alli A. is presently a Computer Science lecturer at Federal University of Agriculture Abeokuta, Nigeria and a Ph. D. candidate at Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He is a registered computer engineer and a chartered information technology practitioner. His main research interests include access control systems, biometrics, image processing and machine learning.

Ogunseye T. recently graduated from the Department of Computer Science, Federal University of Agriculture Abeokuta, Nigeria. He is presently a software developer with an IT firm based in Lagos, Nigeria and his main research interests include access control systems and information security.

How to cite this paper: Adebukola Onashoga, Adebayo Abayomi-Alli, Timileyin Ogunseye,"Enhanced Role Based Access Control Mechanism for Electronic Examination System", IJCNIS, vol.6, no.3, pp.1-9, 2014. DOI: 10.5815/ijcnis.2014.03.01