Role based Access Control with Single Sign-on Architecture using web Services for LMS

Dharmendra Choukse
Institute of Engg. & Science
IPS Academy
Indore, India

Umesh Kumar Singh
Institute of Comp. Science
Vikram University
Ujjain, India

ABSTRACT
In a Modern World, Web Services have been widely used by different industries to improve business functions and productivity, integrate and automate client support, etc. Hence, it is essential to protect the information and all other resources from unauthorised access by controlling the access via a particular system. Nowadays, large organisations are also switching their activities from host-based application platforms to network-distributed, client-server platforms that bring some difficulties for both end-users, who have multiple electronic identities for different systems, and system administrators, who manage multiple applications separately.

Role-Based Access Control (RBAC) is a reasonably novel access control technique that provides a centralised, dynamic, and consistent way to authorise management, specifically for the requirements of a particular industry to improve its security. Since an authentication mechanism is required for personalised, password-protected user accounts, Single Sign-on (SSO) systems can provide authentication across different services. Due to these benefits, SSO is an approach to implement an RBAC enabled system.

This project exploits the RBAC technique and SSO architecture. The objective of this plan is to learn the RBAC technique and SSO approach. The goal is to develop a Web Portal with reusable security and user access control. To achieve this goal, the Web application was designed and implemented. Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network.

Keywords
RBAC,SSO,Webservices

1. INTRODUCTION
The basic idea of RBAC is to give permissions to users indirectly by using roles which are assigned to a particular user. Thus, the user gets a role (or several roles), and then the role (or roles) gives him predefined permissions. The role's indirection is similar to the groups used in UNIX and other operating systems and privilege groupings in database management systems. Groups can only include users as their members. RBAC can hold gatherings of users, authorisations, and other roles "in a single access control model regarding roles and role hierarchies, role activation, and constraints on user/role membership and role set activation" [1].

RBAC controls the users' access to the info and system resources based on users' activities in the system and needs the roles' identification in the system. Such a model is hypothetical to have a set of basic elements such as users, roles, permissions, operations, and objects, as well as relations between these elements [1]. A set of actions and responsibilities related to a particular activity define a role, then permissions to access objects are specified for roles. Afterward, users are assigned to appropriate roles. Organizations may require various numbers of roles and access rules. In most organisations, roles are quite constant while users and the tasks which are assigned to them can be impermanent, and the possibility of reassignment is essential. Therefore, RBAC is a most suitable approach to provide secure association and access to the resources, because "RBAC delivers a powerful mechanism for decreasing the complexity, cost, and potential for a mistake in assigning permissions to users within the association" [2]. Since RBAC has role hierarchies, where a given role can enclose all of the permissions of some other roles, it is the way to go for organisations where roles have overlapping permissions.

Also, RBAC provides authorisation constraints, for the reason that roles must be constrained in their relations to users and permissions, and restrictions are crucial to establishing higher-level access control policies within an organisation [1]. One example of a restriction is the separation of duty, which prevents the possibility of frauds and errors by controlling user-role and role-permission assignments.

Role hierarchies and constraints make RBAC "policy-neutral", so it can suit different access control policies that are beneficial for organisations that have a wide range of security policies or need to modify their access control based on their needs.

NIST RBAC Model

RBAC is known as an "open-ended technology" which can be implemented as simple as well as complex systems. Therefore, it is not meant to be treated as a single model, because such a model would be too thin or too broad, and would stand for one solution out of many. Because of this, the NIST standard which defines the features compulsory for an RBAC system was proposed in [1]. It has two central components: the RBAC Reference Model and the RBAC Functional Specification [1].

The RBAC Reference Model delivers a strict meaning of RBAC sets and relations, and it is intended to define a standard language of terms for use in the prespecified necessities and to set the scope of the RBAC uniformed features. The RBAC Efficient Specification introduces administrative, review, and system functions [1]. Administrative services provide the ability to create, delete, and maintain RBAC elements and relations. Review functions offer the capability to perform query operations on RBAC elements and connections. Lastly, system functions support the management of RBAC attributes on user sessions and making access control decisions.

The NIST RBAC model consists of four model mechanisms: core RBAC, hierarchical RBAC, static separation of duty (SSD) relations, and dynamic separation of duty (DSD) relations (Figure 1). Core RBAC has a nominal set of elements and connections to satisfy the requirements of current RBAC

systems. The components of core RBAC are users, roles, permissions, and sessions. User-role assignment and permission-role assignment are relations which are fundamental in any RBAC system [1] (Figure1). Role activation in core RBAC is a part of the user's session within a computer system.

In core RBAC the administrative functions include adding and deleting users from the set of users and roles from the set of roles; in addition to creating and deleting instances of user-role and permission-to-role assignment. The supporting system functions allow the creation of a user session with a default set of active roles, the addition of an active role to a session, the deletion of a role from a session, and checking if a particular session has permission to perform a request operation on system resources. The review functions permit an administrator to view all of the elements of the model, and their relations, including users, roles, user assignments, role assignments, and session elements [3]. A core RBAC module is required in any RBAC system, but the other modules of the model are free of each other and can be executed separately.

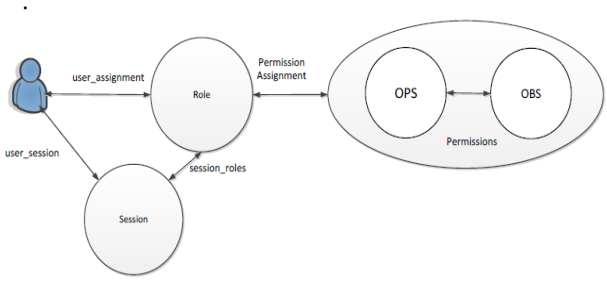

Figure 1: Role-based Access Control Architecture

The Hierarchical RBAC component of NIST (Figure 1) provides relations to support the concept of role hierarchies, i.e. an inheritance relation between roles. Role r1 is a descendant of r2 "only if all permissions of r2 are also permissions of r1, and all users of r1 are also users of r2" [1]. Hierarchical RBAC introduces the concept of a role's set of authorised users and authorised permissions, and defines the two types of role hierarchies: general and limited. General role hierarchies support multiple inheritances, where "a role may inherit permissions from multiple subordinate roles, and more than one role can inherit from the same subordinate role" [3]. Limited role hierarchies enforce role restrictions ensured in a simpler tree structure, namely "a role may have one or more immediate ascendants, but restricted to a single immediate descendant" [3].

The inheritance property significantly simplifies the identification of permission relationships. Roles are allowed permission overlapping, where members of different roles may share permissions. Nowadays, in a corporate environment, many employees share common responsibilities. Role hierarchies are used to avoid multiple role definitions with similar permissions. General role hierarchies are used to deliver practical tools that build rules in such an environment, Limited role hierarchies are more straightforward to implement, but still, provide the hierarchical RBAC functionalities for roles.

Four new administrative functions of hierarchical RBAC extend core RBAC: adding and deleting direct inheritance relationships between two existing roles, as well as creating a new role and adding it as an instant ascendant of a current role and instant successor of a present relationship. The review tasks allow the administrator to view the authorisations and users associated with each role either directly or by inheritance [3].

Constraints in the NIST RBAC model are offered by SSD and DSD components (Figure 1) and used in administrations to enforce conflict of interest policies and prevent users' possibilities to exceed the level of authority for their status. SSD enables exclusive relations between roles according to user assignments, that is, "no user can be concurrently allocated to both roles in SSD" [1]. SSD relations reduce the number of possible permissions to a user by allotment a cardinality constraint on the operators for a set of two or more roles, where cardinality is a number greater than the one identifying a combination of roles that would abuse the SSD policy.

Briefly saying, SSD is a collection of pairs (role set, n), where $n \geq 2$ and no user is assigned to nor more roles from the role set [1]. In SSD, the administrative functions include creating and deleting role sets and adding and deleting role members. The review functions permit an administrator to view the properties of existing SSD sets.

In the same manner, DSD relations (Figure 1) limit the possible permissions available to a user by insertion constraints on the roles that can be activated for a user's sessions. Constraints in DSD are a collection of pairs (role set, n), where $n \geq 2$ and no user session may activate n or more roles from the role set [1]. Also, DSD holds the property of least privilege, where, based on the role being performed, each user has different permissions' levels at different times, and they do not top the time frame that they need to complete the job. The administrative and review functions in DSD are related to SSD's functions, and let an administrator define and view DSD relations.

The main advantage of the NIST standard is that the RBAC System and Managerial Functional Description "provides a functional benchmark for vendors, indicating which abilities must be provided to the user and the general programming interface for those functions" [3]. Also, the specification gives users particular standards for documentation of the requirements and evaluating products, and leaves room to build enhanced features.

SINGLE SIGN-ON

SSO is an access control tool that delivers authentication of a user's access across several software systems and changed services based on the user's permissions while reducing extra logins when the user switches applications within one session [4, 5, 6]. SSO has essential benefits for organisations with standardised infrastructure and centralised users' database where a single user's entity authentication is critical. Different SSO approaches have been introduced by research communities [4, 5], as well as designed and implemented by profitable software makers [9]. Still, it is stimulating to build such a system since there is no standardised way to proceed with SSO application. On one side, each module of the system has to be aware of how to receive and process authenticated calls from a central location; but on the other hand, the central area is hypothetical to know how to map application credentials.

The primary SSO system should contain the user database, an optional cross-reference table, a session control mechanism, and web services, which allow all of the subsystems to obtain standard information. The following subsections will overview two interesting SSO approaches which enrich the basic SSO schema. The first one is the In-VIGO (In-Virtual Information Grid Organization) system [4] with the SSO approach for computational grids, and the second one is the PASS (Privacy-Aware Single Sign-on) system [5] with the enhanced privacy of the users' profiles.

In-VIGO - In-Virtual Information Grid Organization The In-VIGO approach puts three additional layers of virtualisation on the typical grid computing model to hide details of the implementation of lower layers' and allow gridwide processes. The first layer forms pools of virtual resources such as virtual machines, virtual data, virtual applications, and virtual networks to create a virtual computing grid. The second layer encloses services and instances of grid applications, which can be linked as needed to create the virtual information grids.

The third layer creates virtual interfaces to allow different access devices to display combined services, which export their interfaces to users via portals. [4].

Using SSO simplifies and enhances administrative control of all systems as well as users' access across multiple systems, improves network security, and provides an ability to consolidate various systems.

PASS - Privacy-Aware Single Sign-on Nowadays, when an application's personalisation is an essential feature, the implementation of SSO for these systems is increasingly difficult. In this case, user authentication has to be improved by adding personalisation for each application within the SSO network or by providing common personalisation options across the web.

Privacy is a very critical and vital piece of the internet access system which involves storage of the user's information, such as real name and preferences. The SSO system should give the user the ability to specify which properties can be shared between the services and control data exchange, as well as keeping all of the user data confidential from other sources. To keep it all secret, the system should be designed with all of the latest security standards.

Security is a key objective of SSO and should be linked to the entire system by applying the best performs in permission and verification. Although a single point of security failure could be possible in the SSO system, it still gives the ability to manage a user's access to the systems, specifically in RBAC Systems, from one location.

Since privacy is a crucial concern for the PASS protocol, the user is supposed to have the ability to control cross-reference usernames and the distribution of personal data and services. PASS protocol assumes that users and service providers are in a trusted relationship, which is difficult to achieve in real-world internet applications. While most of the web portals are using basic authentication and sending a password as clear text, PASS improves the security of the protocol by issuing the X.509 client certificate based on the new private-public key pair after the initial verification request from the user. "The private key and the certificate are stored on the user device for later reuse" [5].

2. LITERATURE SURVEY

Role-Based Access Control (RBAC) [9] is one of the most important and widely used Web Service access control schemes. In such access control schemes, clients are assigned roles that contain permissions to gain secure access to specific Web Services.

Attribute-Based Access Control (ABAC) models add more dynamicity to the traditional RBAC systems [10].These models make use of attributes owned by the clients, the providers, and some other characteristics related to the environment. Decisions are being made to allow or deny the request based on all these attributes Context-Aware Access Control; RBAC and ABAC access models are providing ways to include contextual information .Other access control models that spotlight on context has been planned as follows Governance Based Access Control (GBAC) [11] The basic idea of the idea is that transactions must be controlled by the relevant legislation to which the organisations sharing the information are answerable. Hence any request for info is verified against the existing laws or guidelines before it is granted the permission.
Location-Based Access Control (LBAC) permits the supplicant to access the resource based on requester's physical location which may be pooled with other attributes related to the identity of the requester. propose merging location with user credentials to support access control decisions.

The Global Roles scheme [12] is one technique where global Web Services rely on global roles. This composition combines more than one local service from different providers. Therefore, the global roles must contain information about all the local services invoked by the worldwide service.
Other systems use policy files, instead of global roles, in both single and composite scenarios to check the validity of the client's request and its possession of the right permissions. Further analysis and operations must be performed in composite Web Services to combine the policy files for all associated services.

Semantic Access Control (SAC) [13] is a new kind of access control model, which uses machine reasoning at a semantic level to determine whether, let the requests pass according to the semantic descriptions of the policies, requests, resources and other entities. SAC more scalable, more applicable to dynamic environments with heterogeneous and complex access criteria. Since the foundation of SAC is the semantic web technologies, it cannot be applied in all access control
Fields.

Shanshan Song and Kai Hwang proposed an enhancing the trust index method of a resource by upgrading its intrusion defence capabilities and also model checks the success rate of jobs on the platforms, but the computing of directed trust is not mentioned in [14].

Wang Meng et al. (2009) [15] proposed a Dynamic Trust Model which is based on recommendation credibility. They suggested a method to differentiate honest and dishonest
Recommendation and adjust the of trust values dynamically. This model defines various participating nodes in the grid as sponsor node, goal node and recommended node.

3. PURPOSE OF THIS PROJECT

The purpose of this plan is to develop a Web Portal with reusable security and user access control. To reach this goal, the Web application will be designed and implemented utilising Façade, Adapter, Data Mapper, and Enterprise Inventory SOA Design Patterns and 3-tier client-server architecture with Data Access, Business Logic, and Presentation layers. A Schedule Application will be implemented as an example of the LMS.

CONSTRUCTING RBAC BASED TEST MODEL

The design of the solution for this project was based on the combination of Role-based Access Control models and SSO architecture principals with industry standards for Web-based Database applications.

The implementation of the project's solution is based on the designed Class Diagram.

Industry standards and common practice techniques for Object-Oriented Applications were utilised for the implementation.

As an example, we implemented a Schedule Application, which demonstrates the beauty of the proposed RBAC system with an SSO approach. This shows how easily the developer can build up an application using third-party controls, and add it to the RBAC Portal. The same technique may be applied to the existing applications by a small authentication modification and perhaps no changes to Role-Base Access if it presents in it.

Flow Diagram

This flow diagram shows the process flow of the authentication mechanism for user login into the Portal. The user must enter a username and password combination and submit the login request. The login process will receive an encrypted password value from the database, and after the decryption, compare it

with the one that the user entered. If the entered combination failed this validation, the user would be prompted to re-enter the login information or recover the password. If the validation succeeded, the user would be redirected to a home page of the Portal, and all of the eligible Applications will be available to the user.

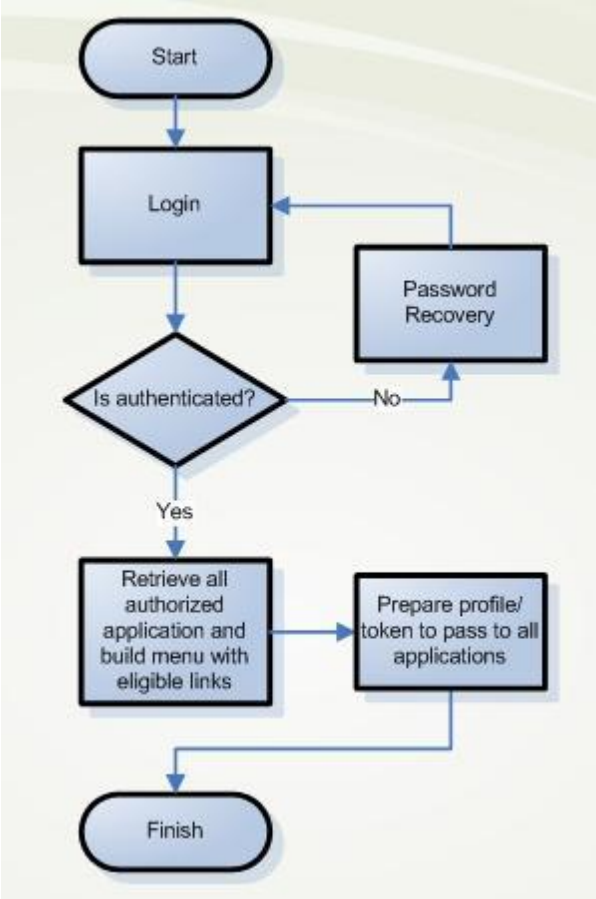

Figure 2. SSO Flow Diagram

Database Design

For portability and extensibility, both the Role-Based management and Single-Sign-On information were separated from other systems at the Database level. For this purpose, I created a stand-alone database named Membership Database which is intended to hold all of the information about users' applications with similar roles. The Scheduling Database was intended to store all of the information related to the Schedule application, which is a sample application for RBAC's implementation for this project. This database stores all of the appointments and resources information. However, the access to the resources will be set up in the Membership Database. Upon entering into the Schedule application, the system will gather all users' credentials from the SSO database via the secure Web Service and then display correct modules within the Schedule.

Membership Database

The Membership Database was designed based on database normalisation rules and Microsoft patterns and practices [10].

The main entity of the Membership Database is the User, which holds all of the required data from the contact information to the last activity tracking. The activity tracking includes failed login attempts and the latest password change. The Application, Module, and Role are other major entities in this database. The Application table holds all of the information about the application, such as location, name, icon, and more. The Module has all of the information for a module with associated Module Items. The relation between the Modules and Application tables is many to one.

The Role is a set of meaningful names within the Application which will be assigned to user-defined Module Items. Module Items is a relation table between the Modules, Item Types, and Permissions. To support the system's flexibility, this design allows the creation of different Item Types to tolerate an add-on SSO application, which specifies particular item types for proper handling. The Status and Status Type allow administering applications/modules/users properly, according to its type. Each entity is assigned to the specific Status, based on which the application can determine if the entity is active, inactive, or has another status. The Session table holds all current sessions and gets processed by the scheduled MS SQL Server task, which removes all of the expired sessions.

This database design allows managing applications with diverse modules and roles and permits distributing the application to different servers/locations within one server room as well as in different places of the world. To be a Portal member, the application must be added to the list of authorised applications of Web Services. Then, the application will be able to verify the user's access and a currently active session and gather other information from the Membership Database. Besides, the application's vendor has to specify the list of the Roles, Modules, and Module Items, which the application will recognise upon receiving it from Web Services.

Web Services

Web Services is a website that is secured by Integrated Windows Authentication. To access this service, the calling application must be identified by a local user. The easiest way to setup the application identification in Windows IIS is to create an Application Pool for each application which runs under a local user identity. The preferred technique is to create separate pools for each application and to assign different users for each one. The main reason for applying this method is the auditing and process tracing.

4. SIMULATION AND RESULT ANALYSIS

The Design of Auto login Process in LMS

The auto-login process begins from a user demand to open the application stimulated when the user clicks the application menu. When the menu has been clicked, the validation of the registration for the login user information will be performed. Once it is complete rightly and properly, the SSO portal will do the hidden background process to open and fill in the form of application login with the login information that has been registered by the user. Subsequently, the portal will show the web frame containing the application opened by the user.

The Design of Auto logout Process in LMS

Another function that has been identified for SSO portal is by automatically logging out the application when the user signs out from the application of SSO portal and then stimulates the function. When logging out, SSO portal will call the logout links of each application managed by the portal. Then, through the portal, it will make a hidden interface that will do a hidden process to log out each application.

Presentation Integration

As explained previously, a user can select the application in secondary domain application through the control panel provided in SSO library management system by the right to use the application.

Figure 3. SSO library management system

To be able to use the application, the user must save the information base by doing registration to be able to access the applications managed by SSO portal.

Figure 4 provides a sample of application registration in the portal, in this case, the webmail of library management system. The user here must give the webmail address and password before confirming and saving the passwords.

Figure 5 above furthermore shows the way to display the application on the navigation menu of SSO library management system. It begins by clicking the edit button on the right side of application that will be displayed. If the status of the application shows that the application is not displayed, the user can display the application by pressing the change button. Once the status of the application has turned into the application is displayed, then clicking the activation of the menu change in the lower part of the application list.

Secondary Domain Application in Portal SSO Frame

To register at the presentation level, SSO portal uses HTML Frame to display the secondary domain application. This domain application will show if the user information base for the intended application is valid.

It is a sample of how the presentation of integration webmail application in SSO portal will be work and presents another example presenting the secondary domain application system for the evaluation of a library management system process.

Table 1 – The Functional Testing of Administrator of SSO library management system.

Functions Tested	Expected Results	Results
Making the SSO portal of the user administrator	The login made can be used to enter the SSO portal	Successful
Deleting the SSO a portal user by administrator	User deleted will be lost including the data of the user	Successful
Seeking the user of SSO portal by administrator	User fulfilling the criteria for searching will display in the table of searching result	Successful
Resetting user's Portal SSO by administrator	The SSO portal user is not able to enter using the old password and must enter using a new password.	Successful

5. CONCLUSION

The objective of this plan was to learn the RBAC technique and SSO approach. The goal was to develop a Web Portal with a reusable security and user access control. I was able to achieve this by improving a centralised Membership Database and Portal Administration application, where the portal administrator and application developers can setup specific security for each application.

As a result of this project, we acquired a better understanding of the Service Oriented Architecture and learned the standard patterns. Also, we was able to apply our knowledge in the designing of a Database and a Class Diagram. Furthermore, we improved my skills in Web

Development and deployment of a Web Application. As an example, we implemented a Schedule Application, which demonstrates the beauty of the proposed RBAC system with an SSO approach. This shows how quickly the developer can build up an application using third-party controls, and add it to the RBAC Portal.

6. REFERENCES

[1] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D.Richard Kuhn, "Proposed NIST standard for role-based access control," ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001.

[2] Gail-Joon Ahn, Ravi Sandhu, "Role-based authorisation constraints specification," ACM Transactions on Information and System Security, Vol. 3, No. 4, November 2000.

[3] William Stallings and Lawrie Brown, "Computer Security: Principles and Practice," Chapter 4: "Access Control," Prentice Hall, August 2007.

[4] Sumalatha Adabala, Andréa Matsunaga, Maurício Tsugawa, Renato Figueiredo, and José A. B. Fortes, "Single sign-on in In-VIGO: role-based access via

delegation mechanisms using short-lived user identities, " In Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium, pages 22b, Santa Fe, New Mexico, April 26-30, 2004

[5] Lars Brückner and Martin Mink, "PASS: A privacy-friendly, secure and open Single Sign-On Protocol for Web Services," Technical Report, Darmstadt University of Technology, IT Transfer Office (ITO), Germany, June 2003

[6] Marek Hatala, Timmy Eap, and Ashok Shah, "Federated security: lightweight security infrastructure for object repositories and Web services," IEEE Conference on Next Generation Web Services Practices (NWeSP'05), pages 287-298, Seoul, Korea, August 23-27, 2005.

[7] Faranak Farzad, Eric Yu, and Patrick C. K. Hung, "Role-based access control requirements model with purpose extension," the 10th Workshop on Requirements Engineering, pages 207- 216, Toronto, Canada, May 17-18, 2007.

[8] Dongwan Shin, Gail-Joon Ahn, Sangrae Cho, and Seunghun Jin, "A role-based infrastructure management system: design and implementation," Concurrency and Computation: Practice & Experience, Vol. 16, No. 11, September 2004.

[9] S. Haibo and H.Fan, "A context-aware role-based access control model for web services," in IEEEInternational Conference on e- Business Engineering, 2005. ICEBE 2005

[10] E. Yuan, J. Tong, B. A. H. Inc, "Attributed based access control for web services," in 2005 IEEE International Conference on Web Services, 2005. ICWS 2005. Proceedings, 2005

[11] R.Joseph manoj, A.Chandrasekar, M.D.Anto Praveena, Gandhi Desai "AFTAC: Attribute, Feedback and Time Decay based Access Control for web services", (ICCCIT 2012)

[12] Cesar Ali "CATRAC: Context Aware Trust and Role based Access Control for composite webservices" 10th IEEE international Conference

[13] X. Wang, J. Luo, A. Song , T. Ma, " Semantic Access Control in Grid Computing". Proc. 11th International Conference on Parallel and Distributed Systems, 2005.on computer and information technology (CIT 2010)

[14] Shanshan Song, Kai Hwang, and Mikin Macwan, "Fuzzy Trust Integration for Security Enforcement in Grid Computing" NPC 2004, LNCS 3222, pp. 9-21.

[15] [Wang Meng; Hongxia Xia; Huazhu Song, "A Dynamic Trust Model Based on Recommendation Credibility in Grid Domain", International Conference CiSE, 2009.