To Study the Assign Authorization to Access the Object through Single Sign-On Web Services in Library Management System (LMS)

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

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

ABSTRACT
In any organization network has a certain way of communication and security based on the network infrastructure. That might support all systems within one physical network containing wireless access, servers, firewalls, access controls and certificates, internal and external devices which enable different subsystems to communicate. The main issue in a large network environment is the importance to distribute the specific individual or group roles to prepare the enterprise for security, and then organize the security by resource and domains, identify the security technologies and complete the requirements to understand how those requirements interact with the network.

Web Services are capable of providing all kinds of services to their clients. The term Web services describe a uniform way of mixing Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol support. XML is used to tag the information, SOAP is used to transfer the information, WSDL is used for relating the facilities are existing and UDDI is used for listing what services are available. Used mainly as a means for businesses to communicate with each other and with consumers, Web services permit organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall. 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
*XML,WSDL,*UDDI,*SOAP,Web Services*

1. INTRODUCTION
One of the main reasons for the use of web services and computer networks are the distribution and access of remote objects. In this context "object" is an abstraction of e.g. files, printers, emails, other resources. Nearly always it is important to protect the objects from illegal use. This problem leads to the identification of users and how they can be authenticated, i.e. proof their identity.

Getting Access to Objects
Before a person can get access to an object, the person needs to identify himself as a user with the permission to access the object. To know what "identify" means, it is important to know what "identity" is:

Identification: Identification is a process in which the user presents an attribute that represents his identity. It is common to use a username for identification.

After presenting the identification attribute to the authority granting access to the objects it protects, the authority has to verify the claimed identity. This process is called "Authentication".

Authentication: Authentication is a process in which the user proves that he is who he claims to be. A usual way to give proof is by presenting a credential, i.e. a shared secret between the authority and the user, e.g. the combination of the username with a password. Only this pair will authenticate the user so he can get the authorization needed to access restricted objects.

Authorization: Authorization gives an identified user the right to access objects depending on his identity.

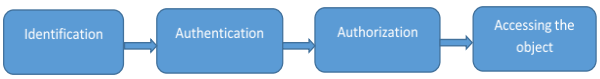

Figure 1. Process for getting access to an object

Figure 1 shows the process which results in getting access to an object. Every authentication authority controls a set of objects (resources) located on machines that are part of the network controlled by the authentication authority. This part of a network is often called a domain or realm. The authentication server assuring that a user's identity is authentic is called domain controller. If the user enters another domain, he needs to re-authenticate in the new domain. If the number and diversity of domains the user have to interact with growing, this will lead to more and more different credentials and different ways to authentication.

SINGLE SIGN-ON
Single Sign-On is a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the necessity to enter multiple passwords." In order to stick to the formerly defined terminology of authentication authorities SSO can be defined as follows:

"Single Sign-On is the ability for a user to authenticate once to a single authentication authority and then access other protected resources without re-authentication" in [].

Although the definition of SSO is quite simple, it is not easy to achieve. There are three primary approaches to get SSO: client based, server-based and service based. The differences between these approaches will be explained in detail as well as the advantages and limitations they bring along in [].

A major problem of single sign-on is the "Key to the Kingdom" problem. The problematic here is that the authentication system makes life easier but it makes security management more

challenging. As the previous sections showed, there is no need for any credentials in a Single Sign-On environment. The "Key to the Kingdom" problem is a synonym for the problem that will arise if the main credential is compromised. This results in the exposure of the whole Single Sign-On environment and all data and resources that are protected by the authentication infrastructure. The same applies to a security breach by poorly developed software in the authentication service (e.g. buffer overflow) which enables an intruder to get authenticated falsely without the use of credentials. Although the SSO environment could mean a higher risk of an attack to intrude the environment, the single sign-on environment itself leads to a solution for that. Because of having only one major authentication authority, the overall security standard of the whole environment is dependent mostly on the security standard of the authentication authority. If the security level of the authentication authority can be improved, it will improve the security for the whole environment. Much expert software developers and web server administrators are specified that the single sign-on protocols that allow workers to sign in to a range of websites with their accounts, suffer from security flaws because of the diversity of Single Sign-On solutions. It is difficult to combine several infrastructures to enhance the range of a Single Sign-On domain.

2. LITERATURE SURVEY

There are a number of researchers have done their work on access control models in web services.

Some of them are as follows.

Bayesian Network based trust and reputation model for web service selection is specified in [1]. This method has three sources for trust calculation such as reputation, QoS monitoring and direct experience of consumer this model, the author tried to overawed some earlier limits by participating the declared bases to find the trust value. The user can require their expectations of services based on QoS, rating mechanism based on consumer feedback on each quality attribute after each transaction, checking whether the feedback is reliable or not and to match the services by finding the similarity of trust rater value and requestor expectation value using the Euclidean method.

Galizia.al. [2] presented a trust model for accessing web service. It follows Trusted Third Party based approach for the classification of the web services with the help of Internet Reasoning Service tool.

Surya Nepal et al. (2010) [3] developed a fuzzy based trust management framework for web service. Originally, they established a data model based on consumer views on QoS attributes that evaluate the reputation of services. Secondly, they proposed the fuzzy-based semantic query model to parse the requested query to evaluate by changed query processing algorithm. They have not lectured some issues such as trust bootstrapping, propagation, retaliation, reciprocation and dishonest or biased ratings.

Priority-based trust (PB) model presented in [4] for service selection in general service-oriented environments. It follows Reputation-based and Trusted Third Party method. It overcomes the boundaries of Certified Reputation Model. PBTrust model is also getting consumer expectation on trust for individual service attribute.

The honest agent can give the feedback and ask other participants in the equal domain about the services. The consistency of the service is calculated as the average of all the feedbacks from contestants [5].

The customer may give the untruthful about the service to make the status value to be reduced. When the trust management center creates this untruthful feedback, the penalty can be given to the customer [6].

Mangling Zhu et al. (2006) [5] designed the social rules on describing the trust relationship between the provider and customer in the open environments. Self Confidence Rule which rates the self-confident of service provider near their providing services. Persistence Rule says that a service provider should be persistent in their goals to achieve better performance. Honest Rule analyzes whether the service provider is trustworthy in their commitments. Motivation Rule checks for motivation in providing services. Reliance Rule estimates the trust from the reliability of service provider. If an agent was unreliable in previous transactions with a consumer, its trustworthiness would be reduced. Reputation Rule discoveries whether it has positive or negative feedback about given that services from the other agents in the open environment. If an agent always performed the committed service, then its reliability will increase, consequently reputation will improve. Trust value of an agent will increase based on their reputation and other dimension and also it automatically updates their reputation. Finally, they defined a trust is based on performance, commitments, social attitude and relations of particulars.

Guha et al. (2003) [7] treated each user as a potential information provider. This model proposed to each user's trustworthiness by broadcasting over a network of people associated with scores or trust.

Cesar Ali et al. (2010) [8] planned a new trust model to access the web services based on context and role of the services requested. Here too they failed to handle new user trust value effectively

Shanshan Song and Kai Hwang proposed an enhancing the trust index method of a resource by upgrading its intrusion defense capabilities and also model checks the success rate of jobs on the platforms, but the computing of directed trust is not mentioned in [9].

Wang Meng et al. (2009) [10] proposed a Dynamic Trust Model which is based on reference credibility. They recommended a method to differentiate honest and dishonest reference and adjust the of trust values dynamically. This model describes various sharing nodes in the grid as sponsor node, goal node and suggested node.

Gao Ying et al. (2010) [11] proposed a trust model based on performance to improve web service security. It is based on the problematic in open service grids to establish trust relationship among dissimilar domains. The authors have planned an algorithm to adjust trust relationships between domains based on entities interactions and also proposed a technique to process recommendation trust.

Kai Wei Shaohua Tang [12] proposed a multi-layer trust computation model based on a direct search in which service providers need to calculate and control the trust of users.

Wu Xiaonian et al. (2009) [13] tried to measure the objects trust according to the entity's performances. This performance trust computation model is based on risk evaluation. This model also features identification of asset, threat and trust

Shashi Bhanwar et al. (2009) [14] planned an access control model based on trust by causal standing and trust of the domain on the basic history of past transactions and rated feedback value.

3. PROBLEM STATEMENT

The major problems identified to access the objects using Single Sign-On web services are mentioned below:

- The process frame to assign the authorization views to the roles is very hard.
- No common mechanism is available to resolve the role conflicts if the user is gaining authorization for permissions associated with conflicting roles.
- How can we impose the restriction based on rules associated with roles?
- Resolve the problem to assign authorization to access the object through Single Sign-On web services.

4. CONSTRUCTING SINGLE SIGN-ON TEST MODEL FOR LMS

In-library management system (LMS), they need users to register for a new account. With the production of web applications, it has become impractical to expect users to remember different usernames and passwords for each application. Web single sign-on (web SSO) protocols allow users to use a single username and password to access different applications. This work examines the web SSO protocol SAML in library management system.

In a library management system (LMS), for example, the set of roles is {student, teacher, director, secretary, admin, borrower, personnel}, the role hierarchy is {<borrower, student>, <borrower, teacher>, <personnel, director>, <personnel, secretary>} (borrower is the super role of student, whereas teacher and personnel is the super-role of director and secretary), SSOD ={<borrower, personnel>, < admin, borrower>}, DSOD={<admin, director>}, the set of objects is {book, borrower Account, personnelAccount}, and the set of activities is {BorrowBook, ReserveBook, GiveBackBook, AdminActivity, ManageAccess, CreateAccount, ModifyAccount, DeliverBook, FixBook}, and the set of contexts is {day(WD), day(HD), day (MD)}, where WD, HD, and MD refer to working day, holiday, and maintenance day, respectively.

In this experiment we have observed applications force users to remember multiple authentication credentials (usernames and passwords) for each application like a set of roles is {student, teacher, director, secretary, admin, borrower, personnel}. Faced with the unreasonable task of memorizing multiple identifications, users re-claim the same passwords, choice weak passwords, or keep a list of all usernames and passwords. Handling multiple verification credentials is annoying for users and weakens security for the authentication system. Web Single Sign-On (Web SSO) systems allow a single username and password to be used for dissimilar web applications. For the user, Web SSO systems help to generate what is called a federated identity. Federated identity managing benefits both the user and the application provider. Users only remember one username and password, so they do not have to suffer from password-amnesia. Application providers also reduce their user management cost. They neither need to support a redundant registration process nor deal with one-time users creating many orphan accounts.

Web SSO systems provide SSO within a single organization. They are called Web Initial Sign-On (WebISO) systems. WebISO systems characteristically provide a web-based module to an organization's current single sign-on arrangement, which presently does not support web-based authentication. Organizations have straight control over their security policies, authentication procedures, and direct access to their user database. Thus, cross-domain issues such as creating trust relationships are not a major concern in WebISO systems.

In addition to federation authentication, federated identity management also involves attribute-based authorization, pseudo-identifiers for privacy, single logout, and IP discovery. Typically most Web SSO protocols address these problems in conjunction with federated authentication. A mechanism for establishing trust relationships between organizations is also covered by SSO protocols.

In-library management system we provide an overview of the Security Assertion Markup Language (SAML) standard as a frame of reference for presenting other standards in internal mechanism of our application. With the help of block diagram, we describe the SAML Web Browser SSO profile in particular. Successful SAML implementations exist in our application called library management system.

The integrated design of the SAML framework allows its components to be combined to support a wide variety of deployment scenarios. SAML consists of core, bindings, and profiles components. Figure 2 shows the relationship between the SAML mechanisms. The profile module defines the context in which SAML is used, and bindings require the protocol used to summarize SAML messages. Bindings and profiles are based on the SAML core, which describes the format of messages and the generic request/response protocols.

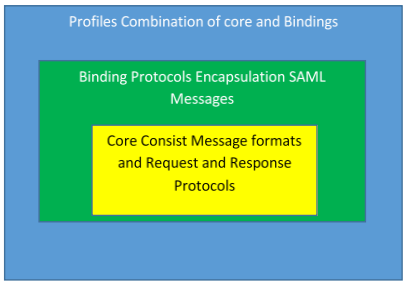

Figure 2: Relationship between profiles, bindings, and the core in the SAML framework.

A. Core

The core consists of security assertions that define the syntax and semantics of messages and general request/response protocols for transferring assertions. Assertions are XML packages that carry SAML statements about the user. For example, authentication assertion may contain statements saying how and when a user was authenticated. The request/response protocol is also specified in XML. In the protocol, a request is a query for an assertion, and a response returns either the assertion or an error. Here, a request for the principal to be authenticated and the response will be an authentication assertion saying whether authentication was successful. The encapsulation of the SAML core, assertions, and protocols, in another common underlying protocol is called a binding.

B. Bindings

SAML bindings specify how SAML protocol messages map to other common protocols such as Simple Object Access Protocol (SOAP) or HTTP. Bindings use standard communications and messaging protocols to allow autonomous SAML-compliant systems to transfer messages securely. Either SAML or the underlying protocol supports mutual authentication, message integrity, and confidentiality. For example, in the SOAP binding, either SOAP or SAML can be secured. XML signatures and encryption are used for application-level security and TLS/SSL for transport layer security. The core and bindings define a SAML use-case called a profile.

C. Profiles

SAML profiles specify how SAML core and bindings are used within a library management system application. Here, the Web Browser SSO profile uses the Authentication request protocol and bindings for HTTP and SOAP. Many types of SAML profiles exist, but SSO specific profiles include the Web Browser SSO Profile, Enhanced Client or Proxy (ECP) Profile,

Identity Provider Discovery Profile, Single Logout Profile, and Name Identifier Management Profile. Although SAML defines many profiles, Web SSO was the primary reason for developing the standard.

A. SAML Web Browser SSO Profile

In-library management system with the help of SAML Web SSO we implemented it in two separate profiles: the LMS Browser Artifact and LMS Brower POST profile. In LMS SAML Web Browser SSO profile was created. The new profile incorporates the two older profiles as bindings. The HTTP Redirect, HTTP POST, and HTTP Artifact in conjunction with the Authentication Request protocol implement the Web Browser SSO profile. Each binding defines a different means of encapsulating the authentication assertions. In particular, Extensible Hypertext Mark-up Language (XHTML) forms transport request/respond messages by value and by reference in the HTTP POST and HTTP Artifact binding respectively.

B. Library Management System Framework for SSO

Although the binding determines the actual messages, the exchange follows a generic model irrespective of the choice of binding. Figure 3 depicts this general exchange pattern.

(1) The UPA attempts to access a resource at the SPE. Assuming the UPA is not authenticated at the SPM, the SPM determines the IPM of the UPA. (The Identity Detection profile can be used to determine the IPM of a UPA.)

(2) The UPA conveys the authentication request message on behalf of the SPM to the IPM.

(3) By some unstated method, the IPM authenticates the UPA.

(4) The IPM responds to the SP by relaying the message through the UPA.

(5) The SPM either grants or denies access to the resource based on the IPM response.

To transfer the message through the UPA to the SP/IP the HTTP POST, HTTP Artifact, or HTTP Redirect binding can be used. However, the HTTP Redirect binding cannot be used in (5) because of the length of the response. For IPM-initiated authentication, the exchange begins at (4).

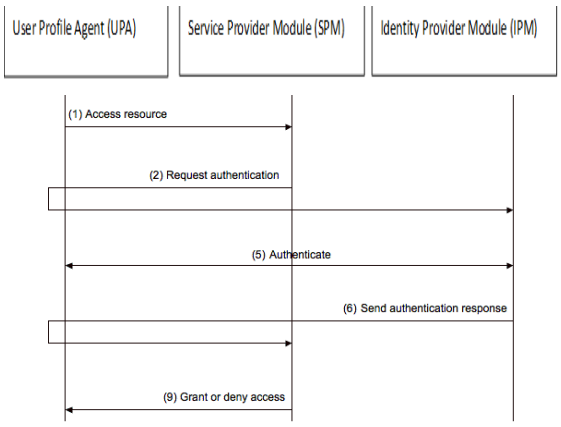

Figure 3: SAML message exchange model for achieving Web SSO.

As a real-world example, in Figure 4, we define the deployment of the Web SSO profile using the HTTP Artifact binding. An artifact is an orientation to a message. An artifact can be resolved to get the content of the message.

(1) The UPA attempts to access a resource at the SPM.

(2) To request SSO service at the IPM, the SPM concerns a request artifact (a reference to the request message) to the ID using the UPA as a middleman.

(3) The IPM asks the SPM to resolve the requested artifact.

(4) The SPM responds with a message containing the original request message.

(5) The ID authenticates the UPA

(6) The ID sends a response artifact to the SPM again using the UPA as a middleman.

(7) The SPM asks the ID to resolve the reaction artifact.

(8) The ID returns the content of the original response.

(9) The SPM grants or denies access to the resource based on the IPM response.

For security, the specification recommends the artifact be transfer to/from the UPA using a secure channel (SSL/TLS) and the SPM and IPM use source authentication before sending the contents of the original message.

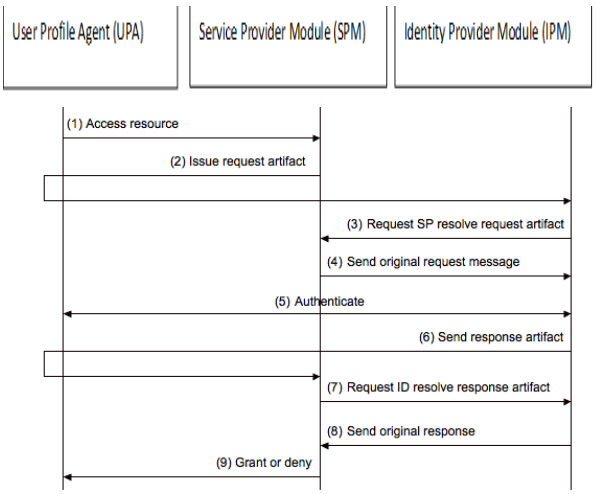

Figure 4: a Specific model for achieving Web SSO using the Artifact binding.

C. 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 explanation of the process above can be modeled in the flowchart in Figure 5 as follows:

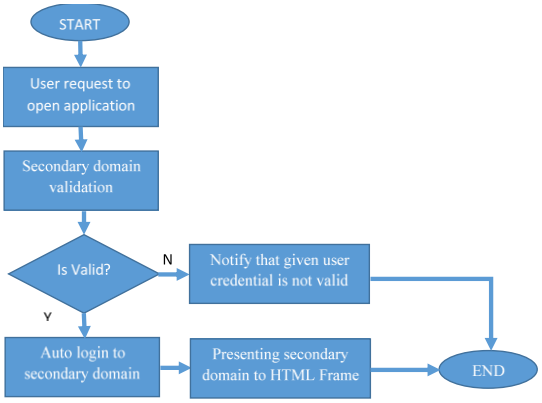

Figure 5: Auto-login Process in LMS

D. 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. The process illustrated above will be shown in Figure 6 below

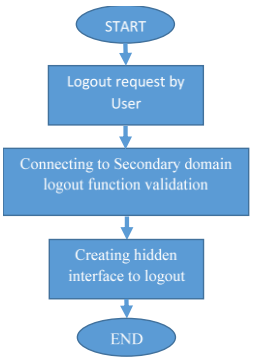

Figure 6: Auto logout process

5. PRESENTATION INTEGRATION AND RESULT ANALYSIS

As explained previously, a user is able to select the application in secondary domain application through the control panel provided in SSO library management system in accordance with the right to use the application. To be able to use the application, the user must save the information base by doing registration in order to be able to access the applications managed by SSO portal. Herewith 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. The 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.

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

Functions Tested	Expected Results	Results
Making the SSO portal of the user by 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

To register at the presentation level, SSO portal uses HTML Frame to display the secondary domain application. This domain application will display 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 sample presenting the secondary domain application system for the evaluation of a library management system process.

Table 2 - below shows the result of the test to the functions of the user existing in the developed Single Sign-On library management system.

Functions Tested	Expected Results	Results
Registration for the application at secondary domain by the user of SSO portal	Information of login user SSO portal in secondary domain the application is stored in the database of SSO portal	Successful
Auto-login to Secondary Domain application	After clicking the link of the menu of secondary domain application, the user of SSO portal is able to auto-login to secondary domain	Successful

	application	
Manage link menu secondary domain an application that will be displayed in navigation menu	In Menu navigation, there will be a link to secondary domain application	Successful
Auto logout from the primary domain	If logging out from SSO portal, it will automatically log out from all secondary domain application	Working in almost all secondary domain application except the application that use dynamic the session key

6. SUMMARY

From the result of the research on the development of single-sign-on portal of library management system, some conclusions are drawn as follows:

1) Using the approach of indirectly single-sign-on developed in portal SSO, the session variables of secondary domain application are still right running well under the session of SS portal.

2) Similarly, the destroy session variable for each of application when logging out from SSO portal mostly can run well; thus enabling to destroy the session in the secondary domain application.

However, for the application of e-learning using the library management system, destroy variable session cannot run well due to the influence of dynamic session key variable from library management system.

3) In view of the use of the frame to do integration presentation, handling the scrolling in the frame is needed. It is caused by the dynamic content of secondary domain application that makes the frame size always be dynamic.

7. REFERENCES

[1] Hien Trang Nguyen, Weiliang Zhao, Jian Yang, "A Trust and Reputation Model Based on Bayesian Network for Web Services", 2010 IEEE International Conference on Web Services

[2] Stefania Galizia, Alessio Gugliotta and John Domingue, A Trust-Based Methodology for Web Service Selection, International Conferences on Semantic Computing, 2007.

[3] Surya Nepal, Wanita Sherchan and Athman Bouguettaya, A Behaviour-Based Trust Model for Service Web, IEEE International Conference on Service Oriented Computing and Applications,2010

[4] Xing Su, Minjie Zhang, Yi Mu, Kwang Mong Sim, PBTrust: A Priority-Based Trust Model for Service Selection in General Service-Oriented Environments, 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing.

[5] Manling Zhu, Lin Liu, Zhi Jin, A Social Trust Model for Services, AWRE 2006 Adelaide, Australia.

[6] Yijiao Zhu a, Junhao Wen a, Mingwen Qin a, Guoli Zhou, Web Service Selection Mechanism with QoS and Trust Management, Journal of Information & Computational Science 8: 12 (2011) 2327– 2334.

[7] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. 2003. Propagation of Trust and Distrust. Proceedings of the 13th Annual International World Wide Web Conference, NewYork,

[8] Cesar Ali "CATRAC: Context-Aware Trust and Role-based Access Control for composite web services" 10th IEEE International Conference on computer and information technology (CIT 2010)

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

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

[11] Gao Ying; Zhan Jiang, "A layered trust model based on behavior in service grid", 2nd International Conference ICACC, 2010.

[12] Kai Wei; Shaohua Tang, "A Multi-level Trust Evaluation Model based on D-S Theory for Grid", International Conference CIS '09. 2009.

[13] Wu Xiaonian; Zhang Runlian; Zhou Shengyuan; Ma Chunbo, "Behavior Trust Computation Model Based on Risk Evaluation in the Grid Environment", WRI World Congress WCSE '09, 2009.

[14] Bhanwar, S.; Bawa, S, "Establishing and Evaluating Trust in a Grid Environment", 10th International symposium ISPAN'09, 2009