Identity Security using Authentication and Authorization in Cloud Computing

D.Ranjith[#1], J.Srinivasan[*2]

[#] *Department of Computer Science and Applications,*
Adhiparasakthi College of Arts and Science,Kalavai,Vellore-632506

Abstract— The Growth of the Cloud Computing invention suggests attractive and innovative computing services through resource pooling and virtualization techniques. Cloud gives delivers different types of Computing Services to their consumers according to their pay per usage of economic model. Anyhow this new technology introduces new immerse for enterprises and bussiness admired their security and privacy. The new cloud services are Security as a service (Saas). This model is mainly for Security enhancemet of a cloud environment. This is a way of gathering security solution under the control of security specialists. Identify and Access control services are the area of security, and sometimes are presented under the term identiy as a service.

Keywords—Cloud Computing,Authentication and Authorization, Security threats.

I. Introduction

Cloud computing, as a new paradigm of information technology, has been developed very quickly in recent years. The vast spread of Internet resources on the web and fast growth of service providers enabled cloud computing systems to become a large scaled IT service model for distributed network environments. Cloud computing is built on top of already existing Internet technologies and is delivered as a self-service utility. Three service models are: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Google, Microsoft Azure Platform, and Amazon Web Services are leading cloud computing vendors in the market of commercial system deployment. Regardless the utilized service model, cloud system can belong to one of the following cloud deployment models: Public, Community, Private or Hybrid[1]. The main characteristics of a cloud environment are abstraction and virtualization which make the technology to be perceived and applied completely in a different manner compared with existing traditional distributed systems. Cloud environment abstracts the implementation details of services and system from users and developers. Besides, resources in cloud computing systems become highly scalable through system virtualization which is achieved by means of resource pooling and sharing [2][3].

Cloud computing has all the security issues associated with distributed applications on the Internet and plus other security issues derived from virtualized and pooled resources[4]. Data storage in a cloud environment is one of the most important concerns from a security point of view. Because multiple cloud customers from the same or different organization can use the same resources or applications, certain security risks should be evaluated and solved before private and sensitive data, applications and system functionality are moved into the cloud. Multi-tenancy requires a policy enforcement mechanism, isolation, service levels, etc. Both cloud deployment model and service model have a high degree of impact on the cloud security solutions and cause different significance on multi-tenancy [5] [6] [7].

Cloud computing security risks depend highly on the cloud service model. IaaS model delivers computing infrastructure, physical storage, and networking as a service. Customers use those resources in order to build their desired computing platforms through platform virtualization facilities.[8] PaaS model adds another layer on the top of IaaS and delivers platform as a service, together with application.

II. Secure Identity management System

A. *Identity Eco System*

In order to support the enhancement of reliable, secure and interoperable identity solutions in an online environment [9]. As mentioned in this Strategy, individuals and private sectors can setup trust relationships between each other in an online environment only based on proper standards for digital identity establishment and authentication. “Identity Ecosystem” eliminates the need for individuals to manage multiple username and passwords for different online services. Individuals with a single digital identity credential can access many different online services, because these service providers trust certain third-party identity providers who manage individuals’ authentication process.The Strategy highlighted four guiding principles about identity solutions in order to have an ideal “Identity Ecosystem”:

- Privacy-enhancing and voluntary identity solutions
- Secure and resilient identity solutions
- Interoperable identity solutions
- Cost-effective and easy to use

Following these principles individuals and private sectors, such as organizations and businesses, will consume interoperable, efficient, easy-to-use and secure identity solutions for online services that will maintain confidence, privacy, choice and innovation. This type of "Identity Ecosystem" will be beneficial for both individuals and private sectors.

B. ICAM Authentication

The Identity, Credential, and Access Management (ICAM) Subcommittee, which is responsible for identity management activities of the US government, has adopted a SAML 2.0 profile which is called ICAM SAML 2.0 Web Browser SSO Profile for supporting and managing proper identity authentication during electronic transactions[10]. The ICAM SAML 2.0 Profile is based on SAML 2.0 specifications provided by the Organization for the Advancement of Structured Information Standards (OASIS). Later in this chapter the SAML standard is shortly introduced. This Profile describes how to facilitate end-user authentication process using SAML message exchange of an identity assertion which carries authentication information in order to support online services.The end-user establishes an identity credential with the IdP in order to request services from the RP. Once authenticated to the IdP, the end-user can access services on the RP site, as it trusts the IdP. Through the end-user activation process the RP manages user's new or existing local account with the identifier obtained from the IdP. Upon receiving a SAML message both entities should verify its digital signature. At the same time, all request-response messages should be verified against metadata.

C.Service-Oriented Architecture

Service-Oriented Architecture (SOA) is an architectural design approach for organizing and using distributed resources which may exist in different business domains. This includes methodologies and rules for designing, developing business solutions and these solutions are delivered as services [11]. SOA provides a frame work for need and capability matching and for uniting capabilities to deal with needs. Basically, interaction is executed through message exchange and the effect is the result of inter action. The main drivers of SOA-based systems are interoperability, usability, scalability and portability [11].

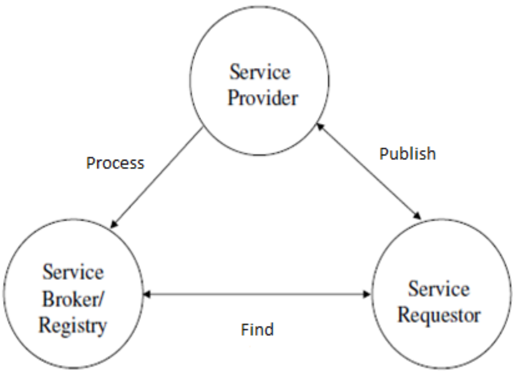

Fig 1. Service oriented Architecture

D.Title and Author Details

According to the Web Service Architecture document, provided by the W3C Working Group, security threats associated with host system, application and network infrastructure are important security considerations for WS environment [12]. Therefore, it is very important to consider them when designing SSO and authorization services.

There are point-to-point and end-to-end security mechanisms and the choice between them is an entirely WS implementation issue. However, point-to point security mechanisms, such as SSL, VPN, IP Sec, etc., do not provide security solutions to ultimate receiver and sender. Because SOAP messages may pass through different intermediaries, end-to-end security technologies are much more appropriate for WS environment. Three security related concepts are important in the WS architecture: resources which should be protected; protection mechanisms (policy enforcement mechanisms), and policy documents which represent constraints on resources. The following are the requirements for assuring end-to-end security in WS environments.

Authentication – One way or in some situations mutual authentication mechanisms should be applied in order to verify the identities of a service provider and a requester.

Authorization – After a successful authentication, an authorization mechanism should control access rights of resource requesters. Role-based access control and policy-based techniques can be used.

There are many Web Service Security related technologies that provide solutions to the above mentioned security problems.

E. SAML Standard

SAML is Security Assertion Markup Language. The purpose of the SAML standard is to describe and exchange security information via SAML assertions between online business domains that trust each other.

This standard has strict syntax and rules for managing SAML assertions. The SAML is the core standard used for designing cloud authentication service in this project and the design is based on cloud authentication frameworks described in the following referenced papers[13][14]. The SSO mechanism benefits from the usage of the SAML standard, which provides a solution to transfer security information independent of any specific platform, domain and protocol. The following subsections briefly introduce SAML standard's main features and rules; complete specifications and details can be found in the OASIS document of SAML V2.0 technical overview [15].

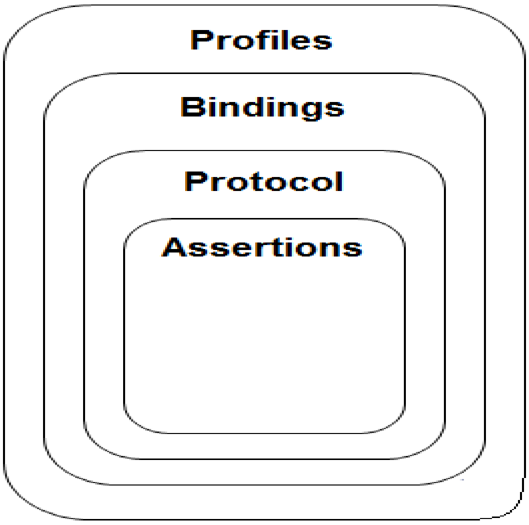

Fig 2. SAML Components

F. Security and Privacy Considerations

XACML-based systems are subject to security and privacy related treats which should be considered during the implementation phase. Because the XACML model is based on interactions and dependencies between the XACML components, it is very important to establish trust relationship between them specific to the concrete system implementation. XACML V3.0 core specification documentation [20] highlights those compromise situations significant for

XACML-enabled environments:

□ Unauthorized disclosure
□ Message replay
□ Message insertion
□ Message deletion
□ Message modification

III . Overview of Cloud Environment Model

The shows logical representation of a cloud environment: different applications running on a cloud platform deliver different services to end-users through the Internet.

End-users can be people or other business entities; however, regardless of the user type, communication channel should be secure. As the picture shows, users interact with an access point entity through the Internet. Here, the access point is the logical representation of a cloud access point acting as an entry point to cloud-based services (practically, there is not only one access point, but application servers run behind different types of access points and proxy servers). As the communication is managed through message exchanges, there may be a need for message encryption, decryption, digital signature, etc., according to enterprise requirements and needs. This means that both communicating parties will need public-private key pairs to protect resources; and as different service providers (enterprise entities) may exist and run their applications within the same cloud environment, appropriate PKI system should be adopted for that environment.

A. Design Approach Security System.

In order to provide security solutions as services, there is a need to define an architecture consisting of entities that handle the security system functionality. Those system entities act as security service providers, which constitute secure environment for cloud-based systems. This research focuses on two types of security services: authentication and authorization. Both services are implemented using Web Service technology and interoperability is one of the main features of the designed security system for delivering those services.

B. Authentication System

A single enterprise may provide many application services to end-users. E-mail servers and web servers are examples of application services providers. As company's boundaries broaden, the number of application services grows. Mostly all service providers should authenticate clients before service transactions are executed, because they are dealing with personal information. This means that the client should have security context for each application server and log in before it can consume any service. The same situation happens when the client accesses resources in different security domains. As mentioned in the second chapter, having many security credentials for authentication purposes is not an effective solution from security, system coordination, and management perspectives. While organizations migrate to cloud environments, the same problem still exists.

C. Authorization System

As already mentioned earlier, different application services may be hosted in a cloud environment and may use the same physical resources. However, each application service is logically separated from others. Different types of system entities consume those services; therefore, application service provider should manage a proper mechanism for access control decisions. This means that various users, after being successfully authenticated, should request and access those resources and services for which they are authorized in a particular enterprise security domain. As the number of the services and service consumers grow, management of access control mechanism becomes more complex and expensive: each service provider needs to implement independent access control mechanism by means of self-governing security policies and policy enforcement points.

IV. DESIGN AND SPECIFICATION OF AUTHENTICATION AND AUTHORIZATION SYSTEMS

This chapter describes the design details of authentication and authorization services and their corresponding communication messages and protocols.

A. *Design of Service Interfaces*

The proposed shared security system is designed using WS technology. All system entities in the shared security system act as service providers and deliver security related solutions to cloud-based entities. They have well defined interfaces which enable service requesters to consume those services without any complexity. Each service has an input parameter (some services may not require an input parameter) and corresponding output parameter. These parameters conform to request and response messages for each service. The request-response messages are wrapped into the XML format, thus making platform independent system entities to interact with one other in an interoperable manner. Particularly, service provider has a description of provided services, which is remotely available to service requesters. In this system, the security service providers register their services and publish their WSDL URLs at the IDMS. The application service provider looks up for the desired service at the IDMS service provider and obtains the URL of the WSDL file for that particular identity service. Then the application service provider obtains the WSDL document and based on that description the service can be easily consumed.

B. *SSO Service*

As described in Chapter 3, SAML server provides a SSO service to application service providers. The end-user authentication process is completely controlled and managed by the central security system of a cloud environment. For this system all SAML messages are transmitted using the HTTP-Redirect or HTTP-POST binding. In order to get a SAML ticket, the PEP server needs to connect to SSO service provider endpoint for incoming requests and call the Request_SAML_Ticket service. Through this call it sends a SAMLAuthenticationRequest message to the SAML server. The message is directed to the SAML server through the central authentication server which acts as a proxy server. The latter intercepts the message. As the request message is for authentication purposes, it starts to authenticate the end-user. The authentication result and SAMLAuthenticationRequest message are passed to the SAML server. In turn, SAML server issues a SAMLAuthenticationResponse message based on the authentication result and request messages. The SAMLAuthenticationRequest message must contain assertion ID for a particular message, ID and service URL of the service requester in this case the ID and service URL of the application service provider. The ID must match the registered ID in the IDSM database and service URL must match the one described in the service metadata. The message also contains assurance level of identity parameters for the authentication process: identity verification will be at that level. There may be other elements included in the request message. At the end, the message should be digitally signed by the service requester. The authentication result contains the subject ID according to the requested format and the status code and value of the identity verification process. The SAMLAuthenticationResponse message must contain assertion ID of the request message, ID and service URL of the SSO service provider, authentication result status, and assurance level of performed identity verification. There may be additional elements included in the response message. Before sending back the response message, it should be digitally signed by the SSO service provider.

C. *SSO Service Protocol*

Any user or client application, before accessing any resource provided by the application service, is first required to be authenticated. The SSO can be IdP-initiated or RP-initiated in our system and the authentication process contains multiple interactions between different system entities. Figure 10 shows communication protocol between participating entities in the SSO service. This is a RP-initiated SSO. The end-user first connects to the application service provider through request resource message in order to request access to a protected resource or service. The request message is intercepted by the PEP server. If the end-user does not have a valid local session for that particular application service, PEP returns an authentication request message, such as SAMLAuthenticationRequest and directs the end-user to the SSO service provider. The user connects to the strong authentication server via HTTP Redirect message protocol. Then the authentication process is executed according to the Strong Authentication Protocol provided by FIPS 196 specification. In the protocol diagram the authentication server authenticates only the end-user and it is based on the user's X.509 certificate. In some cases a user may also authenticate the authentication server. Then the user's identity registration is verified with the IDMS service provider. Besides, the authentication server communicates to the LCA server in order to check the validity of user certificate against certificate revocation list published by the LCA service provider. Getting the certificate verification result, the authentication server requests the SAML server to issue a SAML ticket. The SAMLAuthenticationResponse ticket is returned to the user through the authentication server according to the HTTP Post message protocol. Then the user is redirected to the application service provider. The response message is intercepted by the PEP, which verifies first the ticket validity and then may grant the user access to authorized resources or services based on that SAML ticket. More specifically, if the user has been successfully authenticated, then PEP creates a local valid session.

D. Authorization Service

After successful authentication, the user may request protected resources or services. As described in Chapter 3, central security system is responsible for authorization decisions as well. The PDP server delivers a single service which provides authorization decision based on XACML policies. The service requester (PEP) needs to connect to the PDP service endpoint, obtain a reference to the service object, and call the ***Request_XACML_Authorization_Decision*** service. Through this call PEP server communicates with PDP service provider using authorization decision request and authorization decision response messages, which are in XACML format. As mentioned in Chapter 3, we have adopted SAML-conformant PEP for our designed security system.. Because PDP service provider makes authorization decisions based on policy files in XACML format, there is a need to map each ***SAMLAuthorizationRequest*** message into XACML request context and XACML response context into ***SAMLAuthorizationResponse*** message. In order to map and transfer XACML-formed vrequest-response messages in SAML-based messages, SAML profile of XACML should be used. The complete specification of this profile can be found in the following referenced document [44]. It is the PEP's responsibility to protect resources from incoming requests and initiate an authorization evaluation process. The ***SAMLAuthorizationRequest*** message must contain assertion ID for a particular message, ID and service URL of the application service provider. The ID must match the registered ID in the IDSM database and service URL must match the one described in the service metadata. The message should be digitally signed by the PEP. The ***SAMLAuthorizationResponse*** message must contain assertion ID of the request message, ID and service URL of the PDP service provider. Before sending response message back to the PEP, it should be digitally signed by the PDP service provider.

E.Authorization Service Protocol

PEP must control access to different application services, when user or any other system entity requests access to protected resources or services from the application service provider. Figure 11 shows the communication protocol for the authorization service. The user requests access to a resource at the application service site. PEP server intercepts the request message and constructs a ***SAMLAuthorizationRequest*** message including XACML authorization decision query statement which contains the requested resource URL, action and the role of the user. As it is role-based authorization system, each application service administrator must assign roles to users. As already mentioned in Chapter 3, PAP provides user role assigning service and IDMS provides user attribute retrieving service. In order to obtain user role PEP must query IDMS. ***SAMLAuthorizationRequest*** message is sent to the PDP service provider. Upon receiving the message, the PDP service provider makes the authorization decision for that particular request and returns the XACML authorization decision result back to the PEP in a ***SAMLAuthorizationResponse*** message. The message contains the decision status code and one of the four XACML decision values: Permit, Deny, Indeterminate or Not Applicable. If authorization decision evaluation is successful, meaning that PDP has located the only registered policy for a particular XACML request, then the result contains target rule effect, such as deny or permit, defined by the security administrator. If there is no applicable policy for a particular XACML request (role, resource URL and action), then the result contains a Not Applicable decision value. In case of any XACML authorization decision issuing failure, the service returns the result as indeterminate together with the failure status code. Indeterminate decision value is also returned when PDP has located more than one policy for a particular XACML request. If the same policy contains two identical rules (even if rule effects are different), then the authorization decision result is evaluated against the first encountered target rule. In addition, the response message may contain an obligation or advice element. The PEP enforces the authorization decision: it either permits or denies the access. In case of permit, the application service returns the requested resource.

V. PROTOTYPE IMPLEMENTATION

A prototype implementation of the authorization service based on the proposed design of the authorization solution for a cloud environment. The purpose of the prototype is to demonstrate and test the functionality of the authorization service provider. The implementation of the authorization system consists of two parts: Policy Administration Point (PAP) Service and Policy Decision Point (PDP) Service. The PAP service manages a role-based access control mechanism for security administrators and based on that service, the PDP service manages an authorization service for cloud-based system entities. As described in the previous chapters, the authorization service model is designed completely through the SOA approach, specifically using SOAP-based Web Service technology. Besides, Web Service technology provides all the necessary security mechanisms in order to manage secure environment between service requesters and providers.

A. *Policy Administration*

The implemented PAP service is used by system security administrators for managing and administrating role-based access control policies for their application service environment. It provides user friendly web interface for policy administration which is implemented using Java Server Faces (JSF) Model-View-Controller web framework and deployed in Apache Tomcat and My SQL servers. The Apache Tomcat service provides runtime environment, more specifically, a container for the PAP service provider to deliver policy administration services to security administrators through http/https protocols. This PAP prototype implementation allows only registering one policy per each resource URL. The following subsections describe all functionalities provided by the PAP service.

VI. SYSTEM EVALUATION

The overall evaluation of the proposed security system from two perspectives: integration and security. Integration demonstrates how the proposed security services can be integrated within a cloud environment. Security demonstrates how securely the services are delivered to service requesters.

A. *Web Service Security and Integration Advantages*

Both SSO and authorization services are designed using Web Service technology. As mentioned in the first chapter, cloud computing platform is completely service-oriented and is accessed through high level Web API. That is why the integration of these security services within a cloud environment does not cause technology incompatibility issues. Moreover, it can effectively be deployed and exploited through utilizing all the benefits of service-oriented architecture. Here are the main Web Service advantages that the proposed cloud security system obtains from the service-oriented architectural design:

1. Standardized protocols – Web Service protocol stack comprises four layers: service transport layer, XML messaging layer, service description layer, and service discovery layer. This gives the possibility to choose from broad collection of well defined standard protocols for particular system implantation.

2. Interoperability – the most significant advantage that the security system benefits from the Web Service technology is interoperability. The system delivers its services through public network, such as Internet, in an interoperable manner independent of its implementation platform. Platform-independent service provider and requester communicate with each other without any obstacles.

3. Deployability – security services are deployed over standard Internet technologies on application servers and all incoming and outgoing messages can easily pass through firewalls, using SSL over HTTP channel security mechanism. All these advantages make the system less costly to implement and deploy. The only disadvantage that the Web Service technology may cause to the security system is the stateless interaction between the service requester and provider. That is why additional mechanisms may be required in order to keep track of service requests.

B. *Evaluation of System Security*

Security evaluation is based on the attack-oriented threat model. Threat model gives a formal approach to order potential security issues that makes the system security evaluation easy to understand.. the following passage shows whether both services are protected against those security threats. Replay attacks can be prevented using randomly generated session for both services.

C. *Authentication and Authorization Security Threats and Attacks*

- Network eavesdropping. Listening to network packets and reassembling the messages are sent back and forth between more parties on the network. Although not an attack itself, network eavesdropping can easily interrupt information for its uses in the specific attacks listed in this document.
- Password cracking. If the attacker can't able to establish an anonymous connection with the server, It will try to establish an authenticated connection. So, the attacker must know a valid username and password combination. If you use default account names, you may giving the attacker a head start. Then the attacker only has to crack the account's password. The use of giving weak passwords makes the attacker's job very easier to access.
- Repudiation. The ability of users to deny that they performed specific actions or transactions. Without in sufficient auditing, repudiation attacks are difficult to prove.
- Session hijacking.Middle attacks, session hijacking deceives a server or a client into accepting the upstream host as the actual host. Instead, the upstream host is an attacker's host that is manipulating

the network so the attacker's host appears to be the desired destination.

- Session replay. An attacker steals messages off of the network and replays them in order to steal a user's session.
- Spoofing. An attempt to enter into a system by using a false identity. This can be stolen using credentials or a false IP address. After the attacker successfully enter's as a authenticate user's host elevation of privileges or abuse using authorization can begin.

VII CONCLUSION AND FUTURE WORK

Implementations of cloud security solutions under the concept of Security as a Service are in their awaking phase. It has proposed a cloud security system based on that concept and made contributions in the area of authentication and authorization services for a cloud environment.

Centralization and sharing of those identity services in a separate security infrastructure results in an effective and flexible solution for a cloud environment. This approach enables the entire cloud security system to be controlled and managed much easier, thus raising the quality of provided cloud security solutions. Besides, the system ensures the provisioning of those identity services in a secure and reliable manner.

Through this the solution is provided for building cloud-based identity services, such as authentication and authorization based on the cloud SecaaS model. This solution aims to provide an open and platform-independent architecture of a cloud security system, which is completely service-oriented, thus enabling the system to be scalable, interoperable, loosely coupled and location transparent.

So, According to this cloud security system has been designed for managing authentication and authorization services applying quite new cloud service paradigm, such as Security as a Service. As such, there is a need to do more comprehensive observations and activities within this area and here are some of them. The proposed system supports delivery of only two identity services. Therefore, more identity service features can be added, such as single log out, session refreshment, etc.

ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered.

Causal Productions wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template. To see the list of contributors, please refer to the top of file IEEETran.cls in the IEEE LaTeX distribution.

REFERENCES

[1] National Institute of Standards and Technology (NIST), "The NIST Definition of Cloud Computing." Sep-2011.

[2] B. Sosinsky, Cloud Computing Bible, 1st ed. Wiley, 2011.

[3] P. Kalagiakos and P. Karampelas, "Cloud Computing learning," in 2011 5th International Conference on Application of Information and Communication Technologies (AICT), 2011, pp. 1 –4.

[4] W. Liu, "Research on cloud computing security problem and strategy," in 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012, pp. 1216 –1219.

[5] Cloud Security Alliance, "Top Threats to Cloud Computing V1.0." Mar-2010.

[6] K. Popovic and Z. Hocenski, "Cloud computing security issues and challenges," in 2010 Proceedings of the 33rd International Convention MIPRO, 2010, pp. 344 –349.

[7] X. Tan and B. Ai, "The issues of cloud computing security in high-speed railway," in International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011, vol. 8, pp. 4358 –4363.

[8] J.Srinivasan and D.Ranjith,"Impact of Database Security in Cloud Computing",Proceedings of International Conference on Global Innovations in Technology and Sciences,2013,pp.263-267.

[9] The White House, Washington, "National Strategy for Trusted Identities in Cyberspace." Apr-2011.

[10] Federal Identity, Credentialing, and Access Management, "Security Assertion Markup Language (SAML) 2.0 Web Browser Single Sign-on (SSO) Profile." 16-Dec-2011.

[11] Organization for the Advancement of Structured Information Standards (OASIS), "Reference Model for Service Oriented Architecture v 1.0." 12-Oct-2006.

[12] World Wide Web Consortium (W3C), "Web Services Architecture." 11-Feb-2004.

[13] A. J. Choudhury, P. Kumar, M. Sain, H. Lim, and H. Jae-Lee, "A Strong User Authentication Framework for Cloud Computing," in Services Computing Conference (APSCC), 2011 IEEEAsia-Pacific, 2011, pp. 110 –115.

[14] A. G. Revar and M. D. Bhavsar, "Securing user authentication using single sign-on in Cloud Computing," in 2011 Nirma University International Conference on Engineering (NUiCONE), 2011, pp. 1 –4.

[15] Organization for the Advancement of Structured Information Standards (OASIS), "Security Assertion Markup Language (SAML) V2.0 Technical Overview." 25-Mar-2008.

[16] M. Lorch, D. Kafura, and S. Shah, "An XACML-based policy management and authorization service for globus resources," in Fourth International Workshop on Grid Computing, 2003.Proceedings, 2003, pp. 208 – 210.

[17] G.J. Ahn, H. Hu, J. Lee, and Y. Meng, "Representing and Reasoning about Web Access Control Policies," in Computer Software and Applications Conference (COMPSAC), 2010IEEE 34th Annual, 2010, pp. 137 –146.

[18] A. L. Pereira, "RBAC for High Performance Computing Systems Integration in Grid Computing and Cloud Computing," in 2011 IEEE International Symposium on Parallel andDistributed Processing Workshops and Phd Forum (IPDPSW), 2011, pp. 914 –921.

[19] Organization for the Advancement of Structured Information Standards (OASIS), "Extensible Access Control Markup Language (XACML) Version 3.0." 10-Aug-2010.

[20] Organization for the Advancement of Structured Information Standards (OASIS), "XACML Profile for Role Based Access Control (RBAC)." 13-Feb-2004.