REVIEW OF ACCESS CONTROL MODELS FOR CLOUD COMPUTING

Natarajan Meghanathan

Jackson State University, 1400 Lynch St, Jackson, MS, USA
natarajan.meghanathan@jsums.edu

ABSTRACT

The relationship between users and resources is dynamic in the cloud, and service providers and users are typically not in the same security domain. Identity-based security (e.g., discretionary or mandatory access control models) cannot be used in an open cloud computing environment, where each resource node may not be familiar, or even do not know each other. Users are normally identified by their attributes or characteristics and not by predefined identities. There is often a need for a dynamic access control mechanism to achieve cross-domain authentication. In this paper, we will focus on the following three broad categories of access control models for cloud computing: (1) Role-based models; (2) Attribute-based encryption models and (3) Multi-tenancy models. We will review the existing literature on each of the above access control models and their variants (technical approaches, characteristics, applicability, pros and cons), and identify future research directions for developing access control models for cloud computing environments.

KEYWORDS

Access Control Models, Role-based Access Control, Attribute-based Encryption Model, Multi-tenancy Model, Cloud Computing

1. INTRODUCTION

The three service delivery models for cloud computing are: (1) Software as a Service (SaaS) in which cloud customers use the provider's applications over the Internet; (2) Platform as a Service (PaaS) in which customers deploy their self-created applications on a development platform that a cloud service provider provides; and (3) Infrastructure as a Service (IaaS) in which cloud customers rent processing, storage, network capacity from cloud service provider. The cloud computing paradigm is associated with security concerns both at the providers' end and consumers' end. While providers want to ensure that their resources and services are utilized only by authorized users; consumers would like to ensure that their data is securely maintained in the cloud and that the servers are not compromised.

Access control is a fundamental aspect of information security that is directly tied to the primary characteristics such as confidentiality, integrity and availability. Cloud computing service providers should provide the following basic functionalities from the perspective of access control: (i) Control access to the service features of the cloud based on the specified policies and the level of service purchased by the customer. (ii) Control access to a consumer's data from other consumers in multi-tenant environments. (iii) Control access to both regular user functions

David C. Wyld (Eds) : ICCSEA, SPPR, CSIA, WimoA - 2013
pp. 77–85, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3508

and privileged administrative functions. (iv) Maintain accurate access control policy and up to date user profile information.

Access control models can be traditionally categorized into three types: (1) Discretionary (2) Mandatory and (3) Role-based. In the discretionary access control (DAC) model, the owner of the object decides its access permissions for other users and sets them accordingly. The UNIX operating system is a classical example for discretionary access control model. For example, the subject (i.e., owner of an object) can specify what permissions (read/write/execute) members in the same group may have and also what permissions all others may have. DAC models are usually used only with legacy applications and will incur considerable management overhead in the modern multi-user and multi-application environment, characteristic of distributed systems such as cloud. The Mandatory access control (MAC) models abstract the need for resource-user mapping and hence are more adaptable for distributed systems, compared to DAC models. The MAC model is typically used in multi-level security systems. Here, the access permissions are decided by the administrator of the system, and not by the subject. In a multi-level MAC model, each subject as well as object is identified with a security level of classification (e.g., Unclassified, Classified, Secret and Top Secret). The Bell LaPadula model recommends the "no-read-up" rule and "no-write-down" rule for maintaining *confidentiality* of information. The Biba model recommends the "no-write-up", "no-read-down" and "no-execute-up-or-down" rules for maintaining the integrity of information. In a Role-based access control model (RBAC), a user has access to an object based on his/her assigned role in the system. Roles are defined based on job functions. Permissions are defined on job authority and responsibilities of the job. Operations on the object are invoked based on the permissions. RBAC models are more scalable than the discretionary and mandatory access control models, and more suitable for use in cloud computing environments, especially when the users of the services cannot be tracked with a fixed identity.

The relationship between users and resources is dynamic in the cloud, and service providers and users are typically not in the same security domain. Identity-based security (e.g., discretionary or mandatory access control models) cannot be used in an open cloud computing environment, where each resource node may not be familiar, or even do not know each other. For example, it can be observed that users of a cloud, especially at the SaaS level access the services through the Internet by various means such as mobile phone, notebook or PDA; hence, it is not possible to identify the users by fixed IP addresses. In such situations, one cannot employ the traditional firewalls to filter packets based on fixed IP addresses of users. In a cloud, users are normally identified by their attributes or characteristics and not by predefined identities. Thus, one needs dynamic access control to achieve cross-domain authentication.

In this paper, we will focus on the following three broad categories of access control models for cloud computing: (1) Role-based models; (2) Attribute-based encryption models and (3) Multi-tenancy models. We will review the existing literature on each of the above access control models and their variants (with regards to their technical approaches, characteristics, applicability, pros and cons), and identify future research directions towards developing effective access control models for cloud computing environments.

2. RELATED RESEARCH

For both the grid computing and cloud computing paradigms, there is a common need to be able to define the methods through which consumers discover, request, and use resources provided by third-party central facilities, and also implement highly parallel and distributed computations that execute on these resources. Grids came into existence in the mid 90s to address execution of large scale computation problems on a network of resource-sharing commodity machines that would deliver the same computation power affordable only with expensive supercomputers and large

dedicated clusters at that time. A grid could typically comprise of compute, storage and network resources from multiple geographically distributed organizations, and these resources are normally considered to be heterogeneous with dynamic availability and capacity. The two primary concerns for grid were interoperability and security, as resources come from different administrative domains with varying global and local resource usage policies, as well as different hardware and software configurations and platforms. Most grids employ a batch-scheduled compute model with suitable policies in place to enforce the identification of proper user credentials under which the batch jobs will be run for accounting (e.g., the number of processors needed, duration of allocation, etc) and security purposes.

Condor [1] is a centralized workload management system suited for computation-intensive jobs executed in local closed Grid environments. Its resource management mechanism is similar to that of UNIX (discretionary access control), with some additional modes of access besides the traditional read and write permissions. Legion [2] uses an object-oriented approach wherein all files, services and devices are considered as objects, and are accessed through functions of these objects. Each object can define its own access control policy, typically done using access control list and authentication mechanisms, in a default *MayI* function that is invoked before any other functions of the object may be called. The Globus Grid Toolkit (GT) [3] proposes mechanisms to translate users' grid identities into local identities (which can in turn be verified by the resource providers using appropriate local access control policies) and also allow users' certificates be delegated across many different sites.

With the single sign-on mechanism (e.g., Open Grid Service Infrastructure, OGSI [4]), users can login only once and have access to multiple grid sites, as well as programs can be authorized to access resources on a user's behalf and can further delegate them to other programs. The OGSI operates in conjunction with resource usage brokers (e.g. Gruber [5]) that act as distributed policy enforcement points to enforce both local usage policies and global service level agreements (SLAs) and allow resources at individual sites to be efficiently shared across multiple sites. In [6], the authors propose a flexible attribute-based multi-policy access control (ABMAC) model for grid computing systems in which each autonomous domain may have its own security policy. ABMAC is based on the idea of integrating the individual authorization decisions arrived at for user requests to access resources/services (all of which are identified with their characteristics or attributes) according to the security policy of each domain and arriving at a final decision using a combination algorithm that can be adapted to suit to the resource/operating constraints. The ABMAC approach is more scalable compared to developing a superset of individual domain policies and evaluating user request for resource access according to this superset.

3. ROLE-BASED ACCESS CONTROL MODEL

In a role-based access control (RBAC) model, the role of a user is assigned based on the least privilege concept – i.e. the role with the least amount of permissions or functionalities that is necessary for the job to be done. Task Role-based access control model (TRBAC) [7] has been considered a viable model for cloud computing environments [8] wherein the traditional static access control models such as discretionary, mandatory or simple role-based models cannot be employed. TRBAC can dynamically validate access permissions for users based on the assigned roles and the task the user has to perform with the assigned role. Tasks could be classified as workflow tasks (those that need to be completed in a particular order) that require active access control and non-workflow tasks (those that can be completed in any order) that require passive access control. Workflow tasks driven active role-based access control is time sensitive and the access permissions assigned for users performing these tasks change dynamically with time, depending on the order in which the tasks are to be executed. Care should be taken to ensure that a user has the minimum required privileges to perform a task under a particular role, and that no

role can be assigned to two or more tasks at the same time. Another variant of role-based access control proposed for cloud computing environments is the Attribute-role-based access control (ARBAC) model [9], wherein the data object to be protected are assigned certain attributes and values; a user with a specific role has to submit the appropriate values for these attributes, and are given access to the objects after proper validation by the service provider. A fine-grained key based ARBAC model has been proposed in [10], where users are assigned the private keys or symmetric keys that are used to encrypt/decrypt the values of the attributes defined for the data objects whose privacy needs to be protected.

Bertino et al [11] proposed the temporal-RBAC (TRBAC) model that enables and disables a role at run-time depending on user requests. In [12], the authors argue that in some applications, certain roles need to be static and stay enabled all the time, while it is only the users and permissions that are dynamically assigned. In this context, they proposed a generalized TRBAC (GTRBAC) model that advocates for role activation instead of role enabling. A role is said to be activated if at least one user assumes that role. GTRBAC supports the enabling and disabling of constraints on the maximum active duration allowed to a user and the maximum number of activations of a role by a single user within a particular interval of time. In [13], the authors present an XML-based RBAC policy specification framework to enforce access control in dynamic XML-based web services. However, both GTRBAC and X-RBAC cannot provide trust and context-aware access control (critical for dynamic web services, characteristic of cloud computing environments), and rely solely on identity or capability-based access control. In [14], the authors propose an enhanced hybrid version of the X-RBAC and GTRBAC models, called the X-GTRBAC model. X-GTRBAC relies on the certification provided by trusted third parties (such as any PKI Certification Authority) to assign the roles to users. X-GTRBAC also considers the context (such as time, location, or environmental state at the time the access requests are made) to directly affect the level of trust associated with a user (as part of *user profile*), and incorporates it in its access control decisions. The access privileges for a user/role are based on the threshold (i.e. the trust level) established based on the requestor's access patterns; if the user appears to deviate from his/her usual profile, then the trust level for the user is automatically reduced to prevent potential abuse of privileges. Such a real-time feature of X-GTRBAC suits to the web-based cloud computing environments with diverse customer activity profiles.

4. ATTRIBUTE-BASED ENCRYPTION (ABE) MODEL

Attribute-based encryption (ABE) is more suitable (compared to the traditional public-key infrastructure based or identity-based encryption) to protect the privacy and secrecy of data in a cloud computing environment. ABE is useful when the source of the data knows neither the identity of the recipient nor their public key; but only knows certain attributes of the recipient. For example, imagine user Alice wishing to communicate with her former classmates, but she does not know their email addresses. ABE identifies a user with a set of attributes. In [15], Sahai and Waters (SW) propose ABE as follows: Given a secret key on a set of attributes ω, one can decrypt a ciphertext encrypted with a public key based on a set of attributes ω', only if the sets ω and ω' overlap sufficiently as determined by a threshold value t. The SW scheme also proposes the use of an access tree-based policy to decide on the attributes required to decrypt a message. An example for access tree could be: $Class2005 \wedge (MyCollege \vee MyTeacher)$ implying whichever user who graduated in the class of 2005 either under *MyTeacher* or from *MyCollege* satisfies the policy.

As an extension of the ABE scheme, two variants are proposed in the literature: the Key-Policy based ABE (KP-ABE) scheme and the Ciphertext-Policy based ABE (CP-ABE) scheme. In KP-ABE [16], the ciphertext is associated with a set of attributes and the secret key is associated with the access tree. The encrypting party has no control over who has access to the data and can only

define the set of descriptive attributes necessary to decrypt the ciphertext. There is a trusted authority that generates the secret key, provided the user submits the appropriate values for the attributes that constitute the access tree. In CP-ABE [17], the ciphertext is associated with the access tree and the encrypting party determines the policy under which the data can be decrypted, while the secret key is associated with a set of attributes. In [18], the CP-ABE scheme has been leveraged towards an efficient implementation of the Permission as a Service model to provide users (content owners) with a single point of access control to set permissions on data belonging to multiple services.

A naïve extension of the KP-ABE and CP-ABE schemes for multi-authority systems, characteristic of cloud computing environments, would require each user to hold the attributes or the access tree issued by the different authorities, and there is a need for a global authority that can verify the attributes across different organizations and issue appropriate secret keys to all the users in the system. However, such a global authority is prone to attacks as well as likely to become a bottleneck in an Internet-scale cloud environment. Another major challenge is the possibility of collusion between multiple users (including those whose attributes have been revoked) holding attributes from different authorities to obtain illegal access of data. In [19], the authors have proposed a KDC (Key Distribution Center)-based approach of distributing the decryption key to data owners and users who are assigned a certain set of attributes, which are encrypted along with the data by the owner; users with the matching set of attributes can retrieve the data from the cloud. The attribute-based encryption model applied here is collusion secure as it is based on bilinear pairings on elliptic curves; two users cannot together decode any data that neither of them have individual right to access. The KDC-based access control model is more likely to become a single point of failure (especially when operated with one or fewer KDCs in the cloud), and incurs significant control and management overhead with increase in the number of cloud users and providers.

In [20], the authors propose a multi-authority ABE-based access control model suited for cloud computing environments. According to this scheme, each user is assigned a unique global user identifier (UID) and each user is assigned a unique authority identifier (AID). Both the UID and AID are issued by a certificate authority (CA) trusted by the various authority domains. To prevent two users from colluding together to gain illegal access of data, the CA-certified UID is to be used together with the secret keys issued by different authorities for data decryption. The authors propose an efficient attribute revocation method in multi-authority CP-ABE systems using proxy encryption. The CA-based scheme is more distributed than the KDC-based approach; also a KDC need to be online to distribute the keys for users, whereas a CA need not be online all the time.

5. MULTI-TENANCY MODEL

To be scalable, access control policies need to be defined for groups of VMs that comprise a tenant. Due to the characteristic of sharing of physical resources among tenants whose trustworthiness cannot be easily captured, there is an increased risk of side-channel attacks based on information obtained from physical implementation (e.g., time- or bandwidth-monitoring attacks) [24]. Also, interference of computation from multiple tenants (mainly due to the possibility of existence of covert channels with flawed access control policies) [25] can result in unauthorized information flow on the physical host. A centralized mechanism to globally manage access control can involve a significantly larger number of authorization rules that grows substantially with an increase in the granularity of resources, as well as with the number of users and services supported by the cloud. Today's cloud computing environments demand a varying degree of granularity in the access control mechanisms due to the heterogeneity of services

provided. Thus, there is a need for local autonomy implying that each service model retains administrative control over its resources.

In [21], the authors proposed the separation of security duty between cloud service providers (CSP) and the tenants (customers). They propose a multi-tenant based access control model in which a CSP manages the addition, removal and management of tenants to a cloud and the associated security issues. A tenant in turn manages the access control list of the objects owned by them and the capability list of the subjects belonging to them. For example, in the PaaS cloud model, the CSP should provide a secure computing platform and development environment, whereas customers should assure their applications themselves; in an IaaS model, CSPs should provide trusted infrastructures for customers and customers should secure the rented virtual instances. Recently, Almutairi et al [22] propose distributed security architecture that manifests the above ideas in the form of a trio of virtual resource manager (VRM), access control mechanism (implemented according to the role-based model) and SLA implemented at each layer (SaaS, PaaS and IaaS) of every cloud provider in a multi-tenant multi-cloud environment. Inter cloud operations involving customers at the same layer or different layers as well as intra cloud operations involving customers at the same layer are controlled with this distributed security architecture.

There should be access control policies to facilitate communication between the various tenants of a cloud, especially if they offer services to each other and intend to collaborate over the cloud by taking advantage of the close coupling between the users' machines (i.e., with small latency and large bandwidth). However, it is important to isolate the inter-tenant communication traffic from each other to avoid denial of service attacks. Also, the entity offering the service may want to implement fair sharing of the resources (e.g. bandwidth) among the groups/tenants accessing the service; a group/tenant should not be allowed more than their fair share of service just because they have more machines or better positioned in the cloud network topology. Proper access control policies to rate-limit the tenants should be enforced on clouds that charge tenants on the usage of resources (e.g., bandwidth, number of virtual machines, etc). Also, there should be proper access control policies to protect against floods of authorized traffic between colluders that share a link with the victim. In a virtualized environment, such situations are more common in the clouds, than in the Internet. For example, an attacker could attempt a denial of service attack on a victim virtual machine (VM) *V* by sending a lot of authorized traffic to VM *X* located on the same physical machine with *V*. The idea is to rate limit the allowable traffic based on the destination VMs (rather than the source VMs) running on a hypervisor (for e.g., only 1/*N* of the available bandwidth per VM running on a hypervisor hosting *N* VMs).

In [23], the authors propose a hypervisor-based multi-tenant access control mechanism called CloudPolice and claim that such an approach for cloud access control is more scalable and robust than the typical network-based techniques. Hypervisors have full software programmability, as well as are trusted, network-independent and can block unwanted traffic before reaching the network. To facilitate hypervisor-based access control policies, one could envision several solutions: A naïve solution would be to install all policies and the entire mapping between the active VMs and groups in all hypervisors so that a source hypervisor can directly apply the policy of the destination to all the flows sent by its hosted VMs. However, this approach is not scalable. Another naïve solution is to employ a centralized repository for policies and group membership; hypervisors consult this repository to decide on each new flow and probably cache the access control policies. However, such a centralized service has to sustain very high availability and low response times as well as is likely to be a target for the denial of service attacks. One approach to handle scalability is to have the hypervisors in the cloud to coordinate and push back the rate-limiting/packet dropping filters setup according to the access control policies of the hypervisors of the destination VMs to the hypervisors of the source VMs.

6. FUTURE RESEARCH DIRECTIONS AND CONCLUSIONS

We identify the following future research directions for access control models in cloud computing environments: (1) Develop attribute-driven role-based access control models such that the user-role and role-permission assignments be separately constructed using policies applied on the attributes of users, roles, the objects and the environment; and the attribute-based user-role and role-permission assignment rules be applied in real-time to enforce access control decisions. (2) Develop a location-aware role-based control model incorporated to the Policy Enforcement Point of a cloud (thereby, preventing the disclosure of user's identity, role, or location directly to a remote server in the cloud that may not be fully trusted), and enable/activate the role only when the user is located within the logical positions (computed from real positions by specific mapping functions) that lie within the spatial boundary of a role. (3) Explore software-hardware co-design for security such that the fine-grained access control and usage control mechanisms implemented in software are integrated with new hardware architectural and virtualization features that can help protect the confidentiality and integrity of the data and the resources, even when the powerful underlying hypervisor may be compromised. (4) Mitigate insider threats to the data and resources from the perspective of both a rogue cloud provider administrator and the employee in the victim organization that exploits cloud weaknesses for unauthorized access. (5) Incorporate the relationship between trust and reputation in the access control models for better and secure service quality within the cloud.

The security challenges of cloud computing are exacerbated due to some of its characteristic features such as resource sharing, multi-tenancy and virtualization. Due to the multi-tenancy model of cloud computing, users (tenants) of a cloud computing environment prefer their traffic to be isolated from all other tenants. Though access control for cloud environments are typically provided using techniques such as VLANs and firewalls, these are more suited for enterprise environments and cannot meet the challenges in emerging cloud environments. The challenges include multi-tenancy, diversity in cloud network architectures, scalability (large scale) and the high dynamism of the cloud infrastructure. With multi-tenancy, intra cloud communication (e.g., provider-tenant and tenant-tenant) is becoming a norm and it requires fair sharing between tenants and rate-limiting tenants, which cannot be provided using VLANs and firewalls. In a distributed multi-cloud environment, collaboration between clouds can be either globally federated (consistent with global meta policy), loosely coupled (based on verification of per-cloud access control policies) or ad hoc (establish secure collaboration on a per-user basis). It is possible for all these three collaborations to coexist together in a large scale cloud and systematically update a virtual global directory service on virtualized shareable resources and services of each cloud, manifested across service-level agreements (SLAs).

The new network architectures to evolve for the data centers should employ multiple paths and require specific routing algorithms and address assignments. As today's clouds scale to tens of thousands of physical machines, with a lot more virtual machines added and removed, enterprise-level access control mechanisms will not be scalable enough to handle attacks (e.g., denial of service attacks between cloud tenants) that target a large number of entities, in the order of the magnitude typically seen in the public Internet. Thus, new access control mechanisms for cloud computing environments must be flexible (to support a multi-tenant environment), scalable (handle hundreds of thousands of machines and users) and network independent (decoupled from the underlying network topology, routing and addressing).

REFERENCES

[1] L. Popa, M. Yu, S. Y. Ko, S. Ratnasamy and I. Stoica, "CloudPolice: Taking Access Control out of the Network," Proceedings of the 9th ACM Workshop on Hot Topics in Networks, October 2010.

[2] S. Oh and S. Park, "Task-role-based Access Control Model," Information Systems, vol. 28, no. 6, pp. 533-562, September 2003.

[3] H. A. J. Narayanan and M. H. Gunes, "Ensuring Access Control in Cloud Provisioned Health Care Systems," Proceedings of the IEEE Consumer Communications and Networking Conference, 2011.

[4] S. Sanka, C. Hota and M. Rajarajan, "Secure Data Access in Cloud Computing," Proceedings of the 4th IEEE International Conference on Internet Multimedia Services, December 2010.

[5] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," Proceedings of the 29th IEEE International Conference on Information Communication, pp. 534-542, 2010.

[6] E. E. Mon and T. T. Naing, "The Privacy-aware Access Control System using Attributed-and Role-based Access Control in Private Cloud," Proceedings of the 4th IEEE International Conference on Broadband Network and Multimedia Technology, pp. 447-451, October 2011.

[7] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based Encryption for Fine-Grained Access Control of Encrypted Data," Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89-98, 2006.

[8] J. Bethencourt, A. Sahai and B. Waters, "Cipher text-Policy Attribute-Based Encryption," Proceedings of the IEEE Symposium on Security and Privacy, pp. 321-334, 2007.

[9] K. Yang and X. Jia, "Attribute-based Access Control for Multi-Authority Systems in Cloud Storage," Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems, pp. 536-545, 2012.

[10] T. Ristenpart, E. Tromer, H. Shacham and S. Savage, "Hey, You, Get off my Cloud: Exploring Information Leakage in Third-Party Compute Clouds," Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 199-212, 2009.

[11] D. Nurmi, R. Wolski, C. Grzegorczyk, S. Soman, L. Youseff and D. Zagorodnov, "The Eucalyptus Open-Source Cloud-Computing System," Proceedings of the International Symposium on Cluster Computing and the Grid, pp. 124-131, 2009.

[12] B. Shafiq, J. B. D. Joshi, E. Bertino and A. Ghafoor, "Secure Interoperation in a Multi-domain Environment Employing RBAC Policies," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 11, pp. 1557-1577, Nov. 2005.

[13] A. A. Almutairi, M. I. Sarfraz, S. Basalamah, W. G. Aref and A. Ghafoor, "A Distributed Access Control Architecture for Cloud Computing," IEEE Software, vol. 29, no. 2, March-April 2012.

[14] S. Ruj, A. Nayak and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 91-98, 2011.

[15] K-Y. Chen, C-Y. Lin and T-W. Hou, "The Low-Cost Secure Sessions of Access Control Model for Distributed Applications by Public Personal Smart Cards," Proceedings of the 17th IEEE International Conference on Parallel and Distributed Systems, pp. 894-899, December 2011.

[16] E. Bertino, P. A. Bonatti and E. Ferrari, "TRBAC: A Temporal Role-based Access Control Model," ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 191-233, August 2001.

[17] J. B. D. Joshi, E. Bertino, U. Latif and A. Ghafoor, "A Generalized Temporal Role-Based Access Control Model," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 1, pp. 4-23, January 2005.

[18] R. Bhatti, J. B. D. Joshi, E. Bertino, A. Ghafoor, "Access Control in Dynamic XML-based Web-Services with XRBAC," Proceedings of the 1st International Conference on Web Services, Las Vegas, June 23-26, 2003.

[19] R. Bhatti, E. Bertino and A. Ghafoor, "A Trust-based Context-aware Access Control Model for Web Services," Proceedings of the IEEE International Conference on Web Services (ICWS), pp. 184-191, July 2004.

[20] S. Tuecke, "Open Grid Services Infrastructure," pp. 1-86, www.ggf.org/documents/GFD.15.pdf

[21] C. L. Dumitrescu and I. Foster, "GRUBER: A Grid Resource Usage SLA Broker," Euro-Par 2005, LNCS 3648, pp. 465-474, 2005.

[22] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman, "A Flexible Attribute Based Access Control for Grid Computing," Journal of Grid Computing, vol. 7, no. 2, pp. 169-180, 2009.

[23] H. Jin, W. Qiang, X. Shia nd D. Zou, "RB-GACA: An RBAC Based Grid Access Control Architecture," International Journal of Grid and Utility Computing, vol. 1, no. 1, pp. 61-70, May 2005.

[24] Globus, http://www.globus.org/, last accessed: November 9, 2012.

[25] W. Xiaopeng, L. Junzhou, S. Aibo and M. Teng, "Semantic Access Control in Grid Computing," Proceedings of the 11th International Conference on Parallel and Distributed Systems, vol. 1, pp. 661-667, July 2005.