Design and Evaluation of Policy Based Authorization Model for large scale Distributed Systems

Sarbjeet Singh, Kamalbir Singh and Harpal Kaur

Panjab University, Chandigarh, India

Summary
Large scale distributed systems enable sharing of resources and services scattered over geographically dispersed, heterogeneous, autonomous administrative domains. Two main entities interacting with each other over a distributed system are service requesters and service providers. The service requesters belonging to a particular administrative domain may request access to resources/services available over same or other administrative domains. Similarly a service provider belonging to a particular administrative domain may expose its resources/services over same or other administrative domains. The service requesters belonging to one administrative domain generally have different access rights in different administrative domains. Determining what a service requester is authorized to do in the same or other administrative domains is a difficult task. The overall authorization and access control becomes more complex when service providers attach authorization and access control related policies with their resources/services and provide access to those resources/services based on conformance to established policies. These policies may include authentication, privacy, trust, network workload, business and management etc. related aspects of authorization and access control. Designing an authorization and access control system for such an environment is a complex task and introduces many challenging technology and management related issues. In this paper we have made an attempt to define and implement a policy based authorization and access control framework that can be used to determine the access rights of a subject in different administrative domains and supports policy-based access to resources/services scattered over a distributed system. The framework proposed is scalable, flexible and has been implemented through web services. The paper also discusses prototype implementation of the proposed framework.

Key words:
Distributed Systems, Administrative Domains, Authorization and Access Control, Policy-based Authorization Framework.

1. Introduction

A large scale distributed system is an interconnected set of heterogeneous autonomous systems that cooperatively solve a large problem. The problem is generally divided into a number of independent tasks that are executed in parallel or distributed over different administrative domains over the distributed system for individual processing. These heterogeneous autonomous administrative domains, which are part of the distributed system, use and provide resources that can be shared among different members of the distributed system based on their authorization status and their conformance to established policies. Any administrative domain can have its own set of authentication, privacy, trust, business and management related policies and the domain can also change it at any time [1]. The resources/services have number of policies attached to them and can be accessed by subjects based on their conformance to those policies. Such an environment presents a distinctive set of authorization and access control related challenges that are not addressed by traditional client-server based distributed systems [2].

There are several desirable features that a large scale distributed authorization system must possess in order to be widely usable and acceptable. First, the authorization needs to support multiple security policies and should have the flexibility to allow changes in security policies dynamically [3]. Another desirable feature of authorization system is the support for fine grained access [4]. Support must also be there for a flexible delegation mechanism so that services or resources can be accessed on behalf of a particular user [5], [6]. The authorization system must be context aware also to support and provide context based access. The authorization framework needs to be fully distributed, scalable and manageable also. In this paper we have made an attempt to define a policy based authorization and access control framework that is simple, flexible, scalable, standards based, fully distributed, supports fine grained access to resources/services and is able to enforce local as well as system-wide access control policies.

2. Related Work

A lot of projects like Legion [7], CRISIS [8], CAS [9], PERMIS [10], Akenti [11], PRIMA [12], CARDEA [13], VOMS [14], Shibboleth [15], GridShib [16] and SESAME[17] are developed to address authorization and access control related issues in one or other form. E.g. Legion [7] is an object based distributed computing system. In Legion, there is a layer called “MayI” that

Manuscript received November 5, 2009
Manuscript revised November 20, 2009

decides whether access request should be granted or denied. Each access request passes through it. The "MayI" layer defines a data structure called license that holds authorization related information. The access is decided based on the information present in the license. CRISIS [8] is a wide area security system and defines a new authentication and authorization system. In CRISIS, authentication is based on certificates and authorization is based on ACLs and capabilities i.e. it uses a hybrid approach. The main limitation of Legion and CRISIS is interoperability with web services. These models are not consistent with web services security specifications. CAS [9] is a community authorization service in which service providers delegate the responsibility of maintaining fine grained access to community so it is scalable and supports fine grained access but its approach is somewhat centralized and not truly distributed. Here also the interoperability is the main issue. PERMIS [10] is a policy driven RBAC Privilege Management Infrastructure (PMI). It implements role based access control (RBAC) scheme in which rights are associated with roles rather than with specific entities. PERMIS describes a policy driven role based access control system. The user's roles and the policy are stored in X.509 Attribute Certificates. The policies are written in XML and describe who is trusted to allocate roles to users and what permissions each role has. In PERMIS, policies are written in XML. In our approach, we are using XACML [18] to express policies which is an OASIS standard. Akenti [11] is an access control mechanism that uses digitally-signed certificates to define and enforce an access policy for a set of distributed resources that have multiple, independent and geographically dispersed stakeholders. Akenti [11] allow different stakeholders to express policies specifying how resources can be accessed but the policy language is different from XACML. Akenti policy is expressed in XML and stored in three types of signed certificates: policy certificates, use-condition certificates and attribute certificates. Thus Akenti policy language model is different from XACML policy language model. PRIMA [12] system is particularly motivated by the desire to support spontaneous, short lived collaborations among small group of users. PRIMA provides tools for end users and administrators to manage privileges for the resources they are authoritative for through X.509 Attribute Certificates that carry privilege and policy statements. Cardea [13] is a distributed authorization system that facilitates dynamic access control. It is developed as part of the NASA Information Power Grid, which dynamically evaluates authorization requests according to a set of relevant characteristics of the resource and requester rather than considering specific local identities. In CARDEA, policies are defined with respect to high level identities such as entity's distinguished name. In this, the authorization decisions depend heavily on the attributes a service requester holds. VOMS [14] constitutes a system conceptually similar to CAS. It has a community centric attribute server that issues authorization attributes to members of the community. VOMS uses a format similar to attribute certificates to convey subject attributes. Shibboleth [15] is an attempt to address privacy in an authorization environment and it is primarily focused on using pseudonymity. It is a tool for identity federation between campuses that allows resources to obtain attributes about the user (e.g. departmental affiliation, student status), while preserving the user's privacy and not having to become involved with the details of how the user is authenticated in their home domain. GridShib [16] is an integration of Shibboleth and GT's GSI. It is NSF funded project between NCSA and the University of Chicago. This project will deliver a framework that allows participants in multi-organizational collaborations to control the attribute information that they want to publish, share, and reveal to other parties. Those parties will also be able to determine whether they possess the capabilities to access a service by matching their capabilities with the service's shared policy of required attributes. SESAME [17] is dynamic context-aware access control mechanism for pervasive applications. SESAME complements current authorization mechanisms to dynamically grant and adapt permissions to users based on their current context. The proposed mechanism extends the role based access control (RBAC) model while retaining its advantages. It presents a context-aware access control mechanism but it does not make use of policy language like XACML [18] which is interoperable.

The proposed policy based authorization model is distinguished from these projects/models in one or more of the following ways:

- It supports fine grained access to resources/services through the use of Filter components.
- It uses fully distributed mechanisms.
- Policy expression is platform independent.
- It is able to express and enforce local as well as system wide policies.
- It is flexible, scalable and manageable.
- It uses open standards.

Rest of the paper is organized as follows: Section 3 discusses the elements of the proposed framework. In Section 4 we present the proposed policy based authorization model. Section 5 gives implementation details and results and section 6 concludes the paper with future plans.

3. System Elements

An authorization system can be defined as a system that grants specific type of access to specific requesters based on their authentication, what resources/services they are accessing, current state of the system and their conformation to established policies. It is a detailed description of all aspects of a system dealing with access of resources/services by requesters. In order to understand the proposed model well, we have identified and defined the following elements:

Subject (SU): Subject is an entity that accesses resources/services. It can be a user, a service or any other entity on behalf of user/service.

Service (SR): Service represents specific functionality/feature that can be used/ accessed by Subjects or other Services based on their authorization status and their conformance to established policies. Services are exposed in the environment along with their associated policies and are found by Subjects. Services are provided by different service providers of different administrative domains.

Resource (R): Resource is an object that is used/ accessed by Subjects/Services. It can be a CPU, a storage device, software, data, scientific instrument or any other peripheral. Resources also provide specific functionality/feature. Subjects access Resources through Services. In other words, a Resource is a Service. Like Services, Resources are also accessed by Subjects/Services based on their authorization status and their conformance to established policies.

Service Policy (SrP): Service Policy refers to the set of rules/requirements associated with a Service/Resource. A Subject must conforms to Service Policy in order to access that Service/Resource.

Administrative Domain (AD): Administrative Domain refers to the set of Subjects and Services/Resources under a unique administrative policy. The Services in an Administrative Domain are provided by different Service Providers.

Service Provider (SP): is an entity that exposes Services/Resources in an Administrative Domain. Services/Resources in an Administrative Domain come from different Service Providers under that Domain.

Policy (PD): Policy Database is a repository that stores all the policies of an Administrative Domain. The Policies in a Policy Database are applicable to Subjects/Services/Resources of that Administrative Domain.

Filter: The rights/privileges of a Subject are different in different Administrative Domains. Filter is a component through which access rights/privileges of a Subject are filtered for a particular Administrative Domain. There are two types of Filters (Filter-in and Filter-out). These are explained in section 6.

In a typical large scale distributed system, the elements/entities described above interact with each other in a complex manner. Fig. 1 shows a Distributed System consisting of two Administrative Domains (AD1 and AD2) along with other elements of the system. In the diagram Squares represent Subjects, Diamonds represent Resources, Triangles represent Policies, Rectangles represent Filters and half Ellipses represent Domains. As shown in Fig 1, Subject's access request for Service SR first passes through Filter-out component at the source Domain and then through Filter-in component at the target Domain. During this passage, Subject's access rights are filtered for the target Domain. With Filter-out component, the Subject leaves the source Domain with access rights that source Domain grants to him. With Filter-in component, the Subject enters the target Domain with access rights that target Domain grants to the source Domain. In other words, the Subject gets the intersection of the rights that his source Domain grants to him and the rights that target Domain grants to source Domain. This enables us to implement more fine grained access control in the environment. This also makes the authorization system scalable.

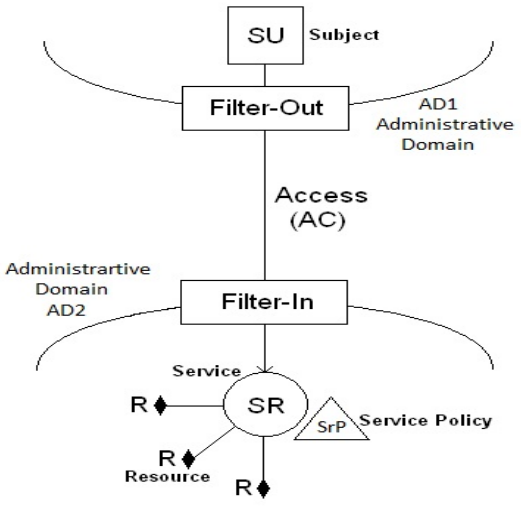

Fig. 1 Distributed System consisting of two Administrative Domains along with other elements of the System.

Fig. 1 also shows Policy database to store different types of policies. These Policies exist in a complex manner among Subjects, Resources and Services of different Administrative Domains. At the target Domain, the proposed Policy-based Authorization Model determines subject's authorization status and checks Subject's conformance to Service Policy. If Subject conforms to Service Policy then target Domain provides the access of requested Resource/Service to Subject, otherwise, the access is denied. Following Section describes the proposed Policy Based Authorization Model in detail.

4. Policy Based Authorization Model

A large scale distributed system may have huge number of subjects and resources/services. Subjects have different roles/privileges in different administrative domains, which results in their different authorization status in different domains. If the number of requesters requesting the services of a target domain is very large then it is very difficult for the target domain to maintain access rights information of all of its requesters. To address this problem, we propose target domain to store access rights information of all of the users/requesters from a source domain as a whole (i.e. the access rights that target domain grants to source domain irrespective of a particular user/requester) and not of the individual users/requesters from the source domain. The source domains will themselves maintain access rights information of their respective subjects. This approach makes the authorization system more scalable. This approach has been implemented in the model using Filter components. There are two types of Filters: Filter-In and Filter-Out. Through Filter-Out component, the subject leaves the source domain with access rights that source domain grants to subject. It attenuates/filters the access rights of a subject for a particular target domain. Through Filter-In component, the subject enters the target domain with access rights that the target domain grants to the source domain. In other words, the subject gets the intersection of the rights that his source domain grants to him and the rights that target domain grants to the source domain [2]. The Filter functionality has been integrated with the XACML based policy model. Fig 2. shows the proposed policy based authorization model. The major components of the model are PEP, PDP and PIP.

As shown in Fig. 2, authorization request from Subject SU is first intercepted by PEP (Policy Enforcement Point). PEP constructs an authorization decision query and passes it to authorization handler. The result of this query determines whether access to Resource/Service is granted or denied. The authorization decision query has details about the identity of the Subjects and the Service requested [2]. Authorization Handler passes this information to PDP (Policy Decision Point). The Policies are retrieved by PDP from Policy Database. The Policies applicable to a resource/service are decided by resource/service owners and are written in Policy Database by policy administrators.

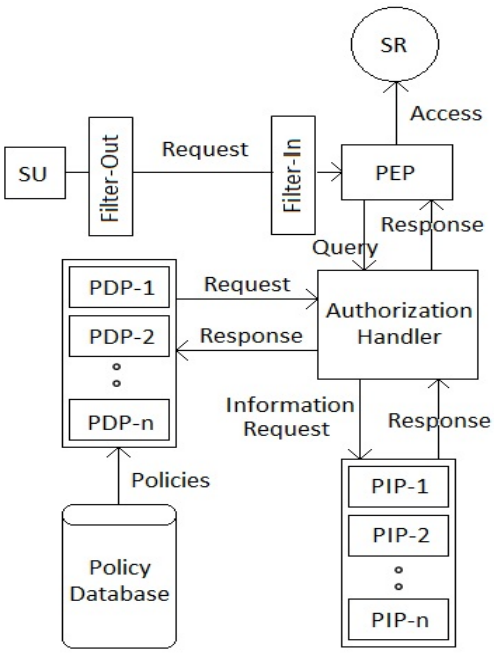

Fig. 2 Policy-based Authorization Model

In the proposed model, PDP has been implemented as a combination of different PDPs (PDP-1, PDP-2 … PDPn). These components (PDP-1, PDP-2, …) implement policy decision functionality specific to a particular technology/mechanism. As the rules to express, store and interpret policies may be different in different systems, we need separate PDPs to implement the PDP functionality of a particular system. E.g. there can be one PDP for access control lists, one for role based access control and another for SAML and XACML based access etc. The Policy Store is capable of importing/exporting policies. PIP (Policy Information Point) is used by Authorization Engine to retrieve Resource, Subject and Environment related attributes. Like PDP, PIP has also been implemented as a combination of different PIPs viz. PIP-1, PIP-2, … PIP-n. As Subject, Resource and Environment attributes are stored in different formats at different places, we need separate PIPs also to fetch, understand and supply those attributes to PEP. For this different PIPs (PIP-1, PIP-2, … ,PIP-n) have been implemented. After getting all the information from PDPs, the Authorization Handler prepares a final result and passes it to PEP. Based on the

result, PEP either grants or denies access to the requested service or resource. Obligation Service, if any, is also executed by PEP.

5. Implementation, Evaluation and Results

The prototype implementation of the proposed model has been done in .NET environment with the support of WSE 3.0 toolkit. WSE 3.0 supports web services security specifications [19] like WS-Security [20], WS-SecureConversation [21] and WS-Trust [22] etc. These specifications addresses security issues like how to associate security tokens with messages (WS-Security), how to request and issue security tokens to establish trust (WS-Trust), how to establish and share security contexts (WS-SecureConversation) etc. These specifications are gaining popularity and becoming standards for handling security requirements of web services. We are making use of these specifications to implement the authorization model described in previous section.

In the prototype implementation we have created 10 Administrative Domains with Subjects ranging from 1 to 15 in each Domain. All the Domains have 1-5 service providers which provide Services/Resources to other Domains. Resources/Services have been exposed as web services. Each Resource/Service has its own Service Policy. Access to a Resource/Service is provided based on conformance to this Service Policy. Policies have been stored in the Policy Database. Policy Database is maintained by every Administrative Domain that stores the policies applicable to Subjects, Services and Resources of that Administrative Domain. The policies have been stored in the database in XACML format. In XACML, Policy is constructed as a set of rules against the target defined as a triod (Subject, Resource, Action).

Fig. 3 shows a high level view of the implementation. As shown in this Figure, Subject's access request for Service SR first passes through Filter-Out component at the source domain and then through Filter-In component at the target domain. During this passage, subject's access rights are filtered for the target domain. At the target domain, PEP prepares authorization decision query and passes it to authorization handler. PEP, PDP and PIP have been implemented as web services. PDP handler gets information from all other PDP implementations (PDP-1, PDP-2 … PDP-n). PDP also makes use of policy database to fetch policies applicable to a particular access request. These policies are evaluated by PDP and evaluation result is prepared. The attributes required for evaluation of policies come to PDP from PIP via authorization handler. PIP obtains attributes related to different aspects using attribute services and passes these to authorization handler. Authorization handler then passes these attributes to PDP for policy evaluation. Thus attributes required for evaluation come to PDP from PIP through authorization handler. This interaction is shown as dotted curved line between PDP handler and PIP handler in Fig. 3. Now PDP prepares final evaluation result and passes it to authorization handler. Authorization handler passes this result to PEP. Based on the response received from authorization handler, PEP either grants or denies access to requested Resource/Service. During all these steps the relevant information is stored in log tables also to address auditing and accounting requirements.

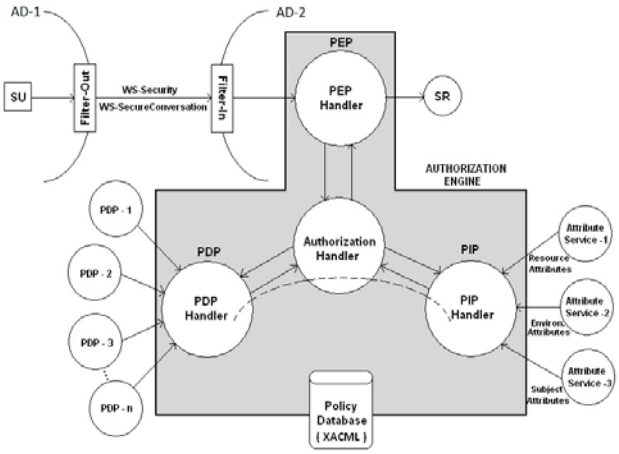

Fig. 3 Schematic showing high level view of the implementation

All information is exchanged as SOAP messages. SOAP messages are constructed using WSE 3.0 toolkit. WS-Security information is embedded while constructing these messages to address encryption and signature requirements. Other web services security specifications like WS-Trust and WS-SecureConversation have also been used for security token exchange and to establish secure communication contexts.

The model has been evaluated by implementing different authorization related scenarios. Fig. 4 presents a general view of the scenarios which have been implemented. The general scenario enables subjects of one administrative domain to access the resources/services of other administrative domains which are protected by security policies and security services of their parent domain. The framework is also capable of implementing different variations of the general scenario.

The performance analysis of authorization policies have also been done. For this, a set of 50 different authorization related policies in XACML have been constructed. These policies have been attached to a sample service one by one and time taken by the different components viz. PIP, PDP

and PEP of authorization handler have been noted. Fig. 5 shows the time taken by the PEP component in evaluating different sets of authorization policies.

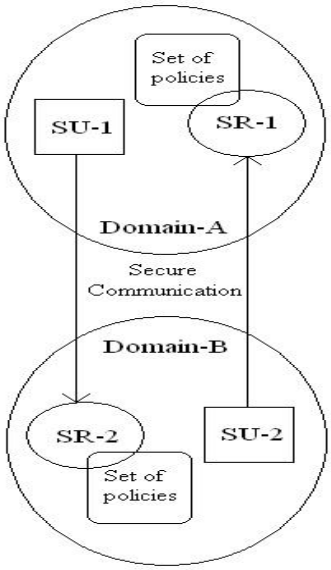

Fig. 4 General view of different authorization related scenarios

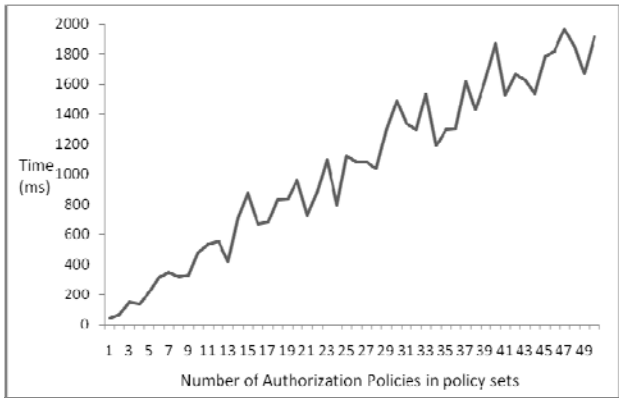

Fig. 5 PEP time to evaluate authorization policy sets of different sizes

The time taken by PEP component to evaluate different authorization policies has also been noted. The details are shown in Fig. 6. The average PEP time comes out to be 72.81 ms which shows that the proposed policy-based authorization model does not add any significant overhead in evaluating different policies applicable to a resource/service. In XACML, the average response time is generally expected to be between 70-100 ms because it involves XML processing. Thus the approach is workable and can be used to provide policy based access to resources/services in a distributed system.

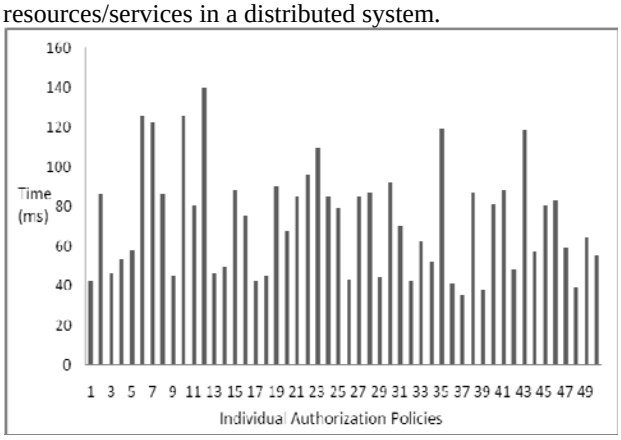

Fig. 6 PEP time to evaluate individual authorization policy

6. Conclusion and Future Scope

The paper proposes a Policy-based authorization model for large scale distributed systems. The proposed model is flexible, scalable, fully distributed, provides fine grained access to resources/services (through Filter components) and is able to express and enforce local as well as system wide policies. Prototype implementation has shown that framework is able to meet identified authorization requirements and supports policy based access to resources/services in a distributed system and thus the approach is workable. Currently, the work on identifying and resolving conflict among different policies is going on. We are also in the process of defining formalized trust and privacy models for distributed systems and integrating these models with the authorization model. After this, a more formal and comprehensive treatment of authorization model is planned.

References

[1] Sarbjeet Singh, Seema Bawa, "Security Policies: Key Factor for Success of Grid Services" at International Conference on Challenges and Opportunities in IT Industry (ICCII) Nov. 2005.

[2] Sarbjeet Singh, "A Security Policy Framework for Grid Services", PhD Thesis, Computer Science & Engineering Department, Thapar University, Patiala, India, July 2009.

[3] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, T. Freeman, A Multipolicy Authorization Framework for Grid Security, Proceedings of the 5th IEEE International Symposium on Network Computing and Applications, pp. 269-272, 2006.

[4] J. Wu, C. B. Leangsuksum, V. Rampure, H. Ong, Policy-based Access Control Framework for Grid Computing, Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid, pp. 391-394, 2006.

[5] L. Seitz, E. Rissanen, T. Sandholm, B. S Firozabadi, O. Mulmo, Policy Administration Control and Delegation

using XACML and Delegent, Proceedings of the Grid Computing Workshop, pp. 49-54, 2005.
[6] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, I. Foster, S. Tuecke, Security Architecture for Open Grid Services, GGF OGSA Security Workgroup, 2003.
[7] http://www.legion.virginia.edu/security_arch.html
[8] E. Belani, A. Vahdat, T. Anderson, M. Dahlin, The CRISIS Wide Area Security Architecture, Proceedings of the 7th USENIX Security Symposium, pp. 15-30, 1998.
[9] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke, A Community Authorization Service for Group Collaboration, Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, pp. 50-59, 2002.
[10] D. W. Chadwick, A. Otenko, E. Ball, Implementing Role Based Access Controls Using X.509 Attribute certificates - the PERMIS Privilege Management Infrastructure, 2002, available at http://sec.isi.salford.ac.uk/download/InternetComputingPaperv4.pdf.
[11] M. Thompson, A. Essiari, S. Mudumbai, Certificate-based Authorization Policy in a PKI Environment, ACM Transactions on Information and System Security, Vol. 6, n. 4, pp. 566-588, 2003.
[12] M. Lorch, D.B. Adams, D. Kafura, M.S.R. Koneni, A. Rathi, S. Shah, The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments, Proceedings of the 4th International Workshop on Grid Computing, pp. 109, 2003.
[13] R. Lepro, Cardea: Dynamic Access Control in Distributed Systems, NAS Technical Report NAS-03-020, 2003.
[14] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell' Agnello, A. Frohner, A. Gianoli, K. Lorentey, F. Spataro, VOMS, an Authorization System for Virtual Organizations, Lecture Notes in Computer Science, Vol. 2970, pp. 33-40, 2004.
[15] Shibboleth Architecture, 2005, available at http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-200509.pdf
[16] V. Welch, T. Barton, K. Keahey, F. Siebenlist, Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration, 4th annual PKI R&D Workshop, 2005, available at http://grid.ncsa.uiuc.edu/papers/gridshib-pki05-final.pdf
[17] G. Zhang, M. Parashar, SESAME: Scalable, Environment Sensitive Access Management Engine, Cluster Computing, Vol. 9, n. 1, pp. 19-27, 2006.
[18] XACML Version 2.0, 2005, available at http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.
[19] Security in a Web Services World: A Proposed Architecture and Roadmap", Joint Security whitepaper from IBM Corporation and Microsoft Corporation, 2002, available at http://www.ibm.com/developerworks/library/specification/ws-secmap.
[20] B. Atkinson, G. Della-Libera, S. Hada, M. Hondo, P. Hallam-Baker, C. Kaler, J. Klein, B. LaMacchia, P. Leach, J. Manferdelli, H. Maruyama, A. Nadalin, N. Nagaratnam, H. Prafullchandra, J. Shewchuk, D. Simon, Web Services Security (WS-Security), 2002, available at http://www.verisign.com/wss/wss.pdf
[21] S. Anderson, J. Bohren, T. Boubez, M. Chanliau, G. Della-Libera, B. Dixon, P. Garg, M. Gudgin, S. Hada, P. Hallam-Baker, M. Hondo, C. Kaler, H. Lockhart, R. Martherus, H. Maruyama, A. Nadalin, N. Nagaratnam, A. Nash, R.Philpott, D. Platt, H. Prafullchandra, M. Sahu, J. Shewchuk, D. Simon, D. Srinivas, E. Waingold, D. Waite, D. Walter, R. Zolfonoon, Web Services Secure Conversation Language (WS-SecureConversation), 2005, available at http://specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf
[22] S. Anderson, J. Bohren, T. Boubez, M. Chanliau, G. Della-Libera, B. Dixon, P. Garg, M. Gudgin, P. Hallam-Baker, M. Hondo, C. Kaler, H. Lockhart, R. Martherus, H. Maruyama, A. Nadalin, N. Nagaratnam, A. Nash, R.Philpott, D. Platt, H. Prafullchandra, M. Sahu, J. Shewchuk, D. Simon, D. Srinivas, E. Waingold, D. Waite, D. Walter, R. Zolfonoon, Web Services Trust Language (WS-Trust), 2005, available at http://specs.xmlsoap.org/ws/2005/ 02/trust/WS-Trust.pdf

Sarbjeet Singh received his B.Tech degree in Computer Science & Engineering from Punjab Technical University, Jallandhar, Punjab, India in 2001 and M.E. degree is Software Engineering from Thapar University, Patiala, India in 2003. He also received Ph.D degree in Computer Science & Engineering from Thapar University, Patiala, India in 2009, working on grid security systems architecture.
Currently he is working as assistant professor in Computer Science & Engineering at UIET, Panjab University, Chandigarh, India. He has more than 10 research publications in international conferences and journals to his credit. His research interests include parallel and distributed systems, distributed security architectures, distributed services like grid and web services, privacy and trust related issues in distributed environments.
Dr. Singh is a life member of Computer Society of India and Indian Society for Technical Education.

Kamalbir Singh received his B.Tech degree in Computer Science & Engineering from Punjab Technical University, Jallandhar, Punjab, India, in 2004. Currently he is pursuing M.E. degree in Computer Science & Engineering from Panjab University, Chandigarh, India. His research interests include distributed systems, distributed policy management and conflict policy identification and resolution mechanisms.

Harpal Kaur received her B.Tech degree in Computer Science & Engineering from Punjab Technical University, Jallandhar, Punjab, India, in 2007. Currently she is pursuing M.E. degree in Computer Science & Engineering from Panjab University, Chandigarh, India. Her research interests include distributed systems, distributed resource management and scheduling algorithms for distributed systems.