A New Trust Framework for Resource-Sharing in the Grid Environment

Hualiang Hu, Deren Chen, and Changqin Huang

College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China
huhualiang@163.net

Abstract. The open and anonymous of grid make the task of controlling access to sharing information more difficult, which cannot be addressed by traditional access control methods. In this paper, we identify access control requirements in such environments and propose a trust based access control framework for grid resource sharing. The framework is an integrated solution involving aspects of trust and recommendation models, based the discretionary access control (DAC), and are applied to grid resource-sharing systems. In this paper, we integrate technology of web services into idea of trust for describing resources.

1 Introduction

Grid applications are distinguished form traditional client-server applications by their simultaneous use of large numbers of resources, dynamic resource requirements, use of resources from multiple administrative domains, complex communication structures, and stringent performance requirements, among others [1].

Although grid security infrastructure (GSI) has been widely adopted as the core component of grid applications, GSI, which provides a basic secure and reliable grid-computing environment, is still at its early stage of development. Since GSI is built upon PKI, risks factors due to the use of PKI have to be considered carefully such as compromising of private keys or theft of certificates for the following reasons:

1. Parallel computations that acquire multiple computational resources introduce the need to establish security relationships not simple between a client and s server, but among potentially hundreds of processes that collectively span many administrative domains.
2. The inter-domain security solutions used for grids must be able to interoperate with, rather than replace, the diverse intra-domain access control technologies inevitably encountered in individual domains.
3. In such a distributed system, a huge set of entities cannot be known in advance.
4. Authentication alone is sometimes not enough to make one confident about allowing a requested access or action rather, a kind of trust is also along with authentication.
5. In order to increase the scalability of a distributed system, it should be possible to delegate the authority to issue access certificates.

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 221 – 228, 2005.

6. In the traditional security systems, an access control mechanism such as Access Control List (ACL) is not expressible and extensible. Whenever new or diverse conditions and restrictions arise, the application is required to change or rebuild [1, 2]. At present, trust for grid is solely built on authentication of identity certificates. As authentication is not insufficient for establishing strong security, it is critical that a proper trust evaluation model for grid is needed. In this paper, we present an access control framework for grid resource-sharing systems, which provides grid users better access control services whilst preserving the decentralized structure of the grid environment. The framework extends a traditional access control model to meet the requirements of grid resource- sharing. The paper is organized as follow. Section 2 identifies the requirements of an access control model for grid environment. Section 3 discusses the characteristics of grid environment. Section 4 explains our access control framework in detail, including the overall architecture, authentication process, scoring scheme. Section 5 gives our concluding remarks.

2 Access Control Requirements

We have identified main requirements that an access control model for grid resource-sharing system should support:

1. Single sign-on: A user should be able to authenticate once.
2. Protection of credentials: User's credentials must be protected.
3. Interoperability with local security solutions: while our security solutions may provide inter-domain access mechanisms, access to local resources will typically be determined by a local security policy that is enforced by local security mechanisms. It is impractical to modify every local resource to accommodate inter-domain. Access; instead, one or more entities in a domain must act as agents of remote clients/users for local resources.
4. Resources may require different authentication and authorization mechanisms and policies, which we will have limited ability to change.
5. Limits are placed on the overall amount of resource consumed by particular groups or users.
6. Decentralized control: The centralized access control authority does not exist in a grid environment. We must take this decentralization of the access control model for grid environment must into account.
7. Node classification: one attribute of grid environment is uncertainty. Their interacting partners are mostly unknown, unlike most other systems, where the users are known. Previously unknown users, who request access to the system, may contact entities. Hence, the framework proposed must provide a mechanism for entities to classify users and assign each user to access rights accordingly.
8. Encourage sharing resources: Sharing is another grid's property. The framework proposed does not only protect the resources, but also provides technologies to grid for resources sharing. This means giving entities the ability to control access to their resources, at the same time, entities must be confident that participation in the system will give them better chance to access to the resources they want.

3 Grid Resource Sharing Systems

Grid resource sharing allows any two nodes in the system to directly access resources from each other systems. To achieve this flexible direct access, grid environment has to support three kinds of operations:

1. Publish: The resources should firstly be described and published in the grid. The provider must pass the identity verification of the resources.
2. Find: There are two ways, which the resources requester finds information. One of them is browse pattern, in the case the result is not unique; the other is drill down pattern, in the case, the result is unique.
3. Bind: By analyzing the binding information gathering from the Services, including the access path of resources, the invoking parameters of resources, the return value, the transmission protocols and the requirements of security; the resource Requester deploys its own system and then invokes the remote resources provided by the resource provider.

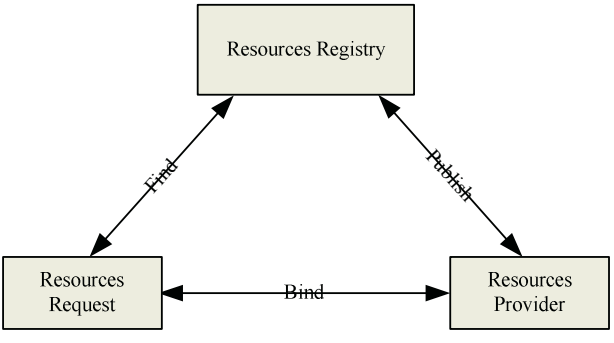

Fig. 1. Relationship of three roles [3]

After three steps, resource is ready. During three steps, there exists following technologies:

1. UDDI (Universal Discovery, Description, Integration). The UDDI specification was first defined by IBM, Microsoft and Ariba in July 2000. It was developed from the DISCO (Discovery of web services) of Microsoft and ADS (Advertisement and Discovery of services) of IBM.UDDI is the registry specification of web services. It defines a way to publish and discover information about web services so that the users who need the service can find and use it conveniently. This specification consists of a core information model provides four aspects of basic information about web services: By means of UDDI, components of web services can complete "register once, access anywhere".
2. WSDL (web service description language). WSDL provides an abstract language for defining the published operations of a service with their respective parameters and data types. It also addresses the definition of the location and binding details of

the service and describes the network into the set of communication end-point, which expresses what a component of web services can do k where it is and how service requester invokes it. Otherwise, WSDL also provides the standard format of describing request for a service requester.

4 Access Control Framework

We propose an access control framework based on the discretionary access control (DAC) model [4]. The basic behind discretionary access control is that the owner of an object should be trusted to manage its security. More specifically, owners are granted full access rights to objects under their control, and are allowed to decide whether access rights to their objects should be passed to other subjects or groups of subjects at their own discretion. In the DAC, a discretional owner of resources has the right for the control of access discretion. Due to anonymousness for grid, we cannot pre-assign access rights to users.

4.1 Terms and Definitions

Following definitions can be found in [5]

Definition 1 (Entities): The entities are all the components involved with the operation of a grid computing system, which can be related to each other via certain trust relationships. These include the user, host (resource provider) node and trusted third parties such as the resource management system (RMS) or a CA.

Definition 2 (Direct Trust): Direct Trust is the belief that one entity holds in another entity in its relevant capacity with reference to a given trust class.

Definition 3 (Indirect Trust): A host node often encounters a client node that it has never met. Therefore, the host has to estimate the client's trustworthiness using recommendations, which a client submits other nodes'.

Definition 4 (Recommended Trust): Recommended Trust expresses the belief in the capacity of an entity to decide whether another entity is reliable in the given trust class and in its honesty when recommending third entities.

Definition 5 (Hard Trust): Hard Trust is trust is the trust derived from cryptographic based mechanisms. This can be treated as a meta-level attribute.

Definition 6 (Soft Trust): Soft Trust is the trust relationship derives from non-cryptographic based mechanisms which employ methods like recommendation protocol, observations, interactions or combination of them.

In this paper, we only discuss Soft Trust, which can address the uncertainty for grid.

4.2 Overall Architecture

Our framework contains client (user) node, service provider node and trust third parties such as the resource management system (RMS) or a CA. Shared resources in framework are rated depending on their size and content; each resource being assigned two

thresholds which capture two access aspects. Only if the request end's two access values both equal to and greater than the corresponding thresholds of the resource. It can access resources that it wants. The request end is responsible to collect recommendations that contain the information needed to evaluate its access values for a particular host. After each transaction, direct trust and direct contribution of both the client node and host nodes are updated accordingly to the satisfaction level of the transaction, which then affect the future evaluation of the access values between these two nodes.

4.3 Authentication

An issue in determining trust is how the user authenticated to the current session. Hence, the authentication, in our framework, must be mutual. This means both a client node and a host node need to be authenticated with each other. The authentication process is initialized by the client node that wishes to make contact.

A user may have an X.509 certificate that he uses from certain machines, a password that he may use from other sites, and may not even want to type in a password from a very un-trusted site. The user may also not yet have some of the authentication methods available [7].

The host node' local database tracks client nodes' records, for authentication purposes and access control purposes such as trust and contribution calculation. In summary, the authentication information that is saved per session consists of a 128-bit GUID (global unique identifier [8]) number and a pair of public/private keys. The node gives out the GUID and the public key as its identity and uses the private key for authentication, the name (nickname) of the user, the authentication method (certificate, password, null), and fields that other users might have added after the user authenticated into the session.

4.4 Evaluating System

A host node classifies its client nodes based on their records in grid environment. After authentication process with the host, a client node is required to send its rating certificates to the host to calculate the client's relative access values. The host node perceives the client's trustworthiness and contribution level, based on the client's relative access values (trust value and contribution value). The trust value is to ensure the node is trusted to interact with. The contribution value is to promote contribution for the host in grid environment.

In our framework, the trust value is key factors for making access control decision can address uncertainty factor in grid. By the trust value, a host node can determine whether it should trust that client to allow it to access to the local resources.

4.5 Rating Certificate Management

A rating certificate contains the direct trust value and the direct contribution score of the recommending node on the recommended node and contains the new trust value and the updated contribution score. After transaction, system automatically updates trust value and contribution value.

4.6 Interaction Procedure

We introduce definition for describing entities' interaction procedure [6]. The interaction procedure is shown as figure 2.

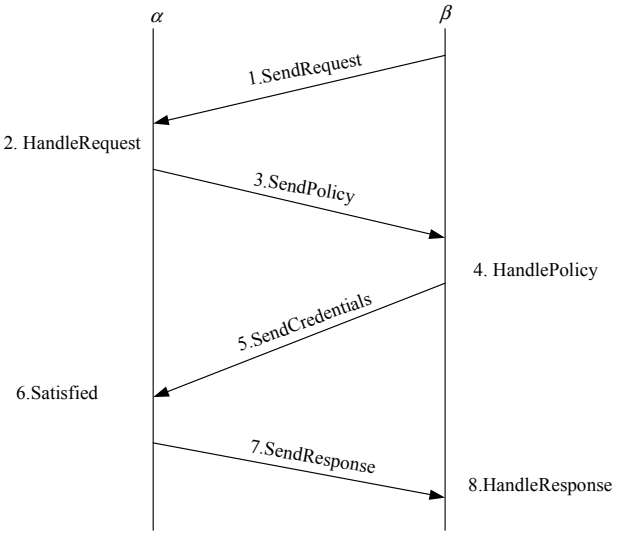

Fig. 2. Interact Procedure

Entities include α and β.

Definition 1 α.Trust(β): α trustsβ;

Definition 2 α.MakeCommentβ): αcomments on β;

Definition 3 α.SendRequest(β): α sends an request to β;

Definition 4 α.HandleRequest(β.Request): αhandles with request which β sends to α;

Definition 5 α.SendPolicy(β): αsends policy toβ;

Definition 6 α.SendPolicy(β. Policy): αhandles with β'policy;

Definition 7 α.SendCredentials(β): αsends Credentials to β;

Definition 8 α.Satistied(β.Credentials): αhandles withβ's message for credentials, to judge whether ß satisfies à'trust-policy;

Definition 9 α.SendResponse(β): αsends the result to β;

Definition 10 α.HandleResponse(β.Response): αhandles with the result for β.Response;

5 Relational Works

In this section, we examine several papers that examine issues that are peripherally related. A model for supposed in [9]. This trust-based model allows entities to tune their understanding of another entity's recommendations. A model for supporting

behavior trust based on experience and reputation is proposed in [10]. The Community Authorization Service (CAS) in [11]. A design and implementation of a secure service discovery service (SDS) is in [12]. The SDS can be used by service providers as well as clients. Formalizations of trust in computing systems were done by Marsh [13]. He attempted to integrate the various facets of trust from the disciplines of economics, psychology, philosophy and sociology.

6 Concluding Remarks

We propose a novel trust framework based on the DAC model for resources-sharing in grid environment.

Our proposed trust model is optimistic in that the majority of nodes start with access value, while a small group of nodes have overdue access value. The concept of trust value can address uncertainty problem. Hence, host nodes can assigns appropriate access privileges to each visitor accordingly.

The trust framework we proposed can also provide trust guarantees to each node. We provided rating certificate that stores and updates the trust value and the contribution value. Involvement values, which are associated with each community within which a node is a member, play an important role in determining accurate trust value of nodes.

Finally we explained the interact procedure between entities.

Our future challenge ahead is to refinement the proposed framework based access scheme in grid environment and how to compute trust value efficiently over the grid.

References

1. I. Foster and C. Kesselman. A Security Architecture for Computational Grids. In Proceeding of the 5th ACM Conference on Computer and Communication Security, November 2-5, 1998.
2. H. Lei, G. C. Shoja. Dynamic Distributed Trust Model to Control Access to Resources over the Internet, Communications, Computers and Signal Processing. August 28-30, 2003.
3. Dr. Liang-Jie (L.J) Zhang. On Demand Business Collaboration With Services Computing.
4. J. McLean. The Specification and Modeling of Computer Security. IEEE Computer, Jan 1990.
5. L. Ching, V. Varadharajan and W. Yan. Enhancing Grid Security With Trust Management. In proceedings of the 2004 (IEEE) international Conference on Services Computing (SCC 04).
6. N.YuPeng and C. YuanDa. Automation Trust Establishment in Open Network Environment. Computer Engineering 2004; 30(16);124-125.
7. D. Agarwal, M. Thompson, M. Perry and M. lorch, A New Security Model for Collaborative Environments.http://dsd.lbl.gov/Collaboratories/Publications/WACE-IncTrust-final-2003.pdf.
8. D. Box, Essential COM, Addison Wesley, 1998.
9. Karl Aberer & Zoran Despotovic, Managing Trust in a Node-2-Node Information System, ACM CIKM, Nov 2001.

10. A. Abdul-Rahman and S. Hail. Supporting Trust in Virtual Communities. Hawaii Int'I Conference on System Sciences, 2000.
11. L. Pearlman et al. A Community Authorization Service for Group Collaboration. IEEE Workshop on Policies for Distributed Systems and Networks, 2002.
12. S. E. Czerwinski, B. Y. Zhao et al. An Architecture for a Secure Service Discovery Service. 5th Annual Int'I Conference on Mobile Computing and Networks(Mobicom'99).
13. S. Marsh. Formalizing Trust as a Computational Concept. Ph.D. Thesis, University of Stirling, 1994.