

Access Control Strategies for Virtualized Environments in Grid Computing Systems

Anna Cinzia Squicciarini, Elisa Bertino
*Computer Science Department
Purdue University
{squiccia, bertino}@cs.purdue.edu*
Sebastien Goasguen
*Computer Science Department
Clemson University
sebgoa@clemson.edu*

Abstract

The development of adequate security solutions and in particular of authentication and authorization techniques for grid computing systems is a challenging task. Challenges arise from the heterogeneity of users, the presence of multiple security administration entities, the heterogeneity of security techniques used at the various grid hosts, the scalability requirements, and the need for high-level policies concerning resource sharing. Recent trends, like accessing grid through science gateways and the use of virtual organizations (VO) for managing user communities, further complicate the problem of security for grid computing systems. Currently, the GSI component developed as part of the Globus Toolkit, the de-facto standard for grid infrastructures is not fully suited to meet those challenges. The main drawback of such an approach is that it relies on a low level identity-based authorization scheme. A low-level access control policy maps a user's identity (distinguished name) to a local account. Such approach does not scale and does not address many of the outlined requirements. We thus need security solutions that go beyond the simple solutions currently in use. The goal of this paper is to make a first step towards such solutions. The paper discusses and analyzes authentication and authorization solutions that better fit novel grid infrastructures characterized by virtual organizations and science gateways. Some of these solutions derive from ongoing work in current grid infrastructure projects; others are new proposals that we think worthy of discussion. In particular, we analyze the solutions proposed as part of the GridShib and the VO Privilege projects

1. Introduction

Grid computing [15, 16] represents an important infrastructure that makes it possible for multiple institutions to pool their computing resources and to collaborate in order to solve computationally intensive problems. As more organizations and users are interested in using grid computing systems in a variety of application domains, security is a key issue. Developing security approaches suitable for such a context requires addressing several requirements, such as interoperability of security mechanisms, authentication, authorization and scalability. Grid sites must be able to interoperate while continuing to use their local security solutions. Achieving interoperability is a complex task; it may require interoperability among the security mechanisms as well as the coordination of access control and authorization policies. To this extent, a unified fine-grained access control mechanism should be developed, based not only on local user identities but also on other qualifying user attributes. Such a requirement is important in order to provide high-level access control policies that can be easily specified and understood. Grid systems dynamically evolve over time. Users and resources can be dynamically added/removed as specific projects are started/completed. Security mechanisms should be designed to reduce the security administrative overhead when dealing with re-configurations of the grid system, the user communities, and the available resources. Finally, establishing and managing trust relationship in a grid-computing environment is still an open problem.

Although authorization in distributed systems has been extensively investigated, not much work has been carried out to address authorization problems in real

large distributed systems such as grids. The current de-facto solution, represented by the GSI component [7] included in the Globus toolkit [2], adopts a simple low-level approach to authorization. Such approach relies on an access control list (*gridmapfile*) that maps a user's global identity (distinguished name, or DN) to a local account. Users whose DN appears on such list are authorized to use the resources, with privileges associated with the local account. This simple approach is in essence the same authorization mechanism used for a single machine, like the Unix mechanism based on the "/etc/passwd" file. In a distributed system like a grid, there may be thousands of users and it is not realistic to base authorization decisions on individual users' identities in that maintaining a grid-mapfile with thousands of entries does not appear viable. Instead, an attribute-based authorization system is desirable.

In attribute-based authorization, the access control policy does not mention individual user's identity, but rather attributes such as the ones describing a person's role in an organization. It has been widely acknowledged in the grid-computing community that attribute-based authorization should be the direction of development for grid security. There are a number of projects investigating such approach, such as the VO Privilege Project [10], GridShib [4], and PERMIS [13]. However, there are quite a few decision dimensions when it comes about designing an attribute-based authorization scheme for grid computing. A suitable approach should address the emerging trends in grid computing. In particular, we observe that recent years have seen grid computing moving towards "virtualized" environments. In such environments, the usage of computational resources is not any longer delimited by institutional boundaries, and many users subscribe to virtual communities whose members have similar interests or computational needs. Such communities often transcend institutional boundaries, and many of them are represented by subscribers to particular websites that serve as a front-end for high-performance computing. These trends are also sometimes demonstrated by terminology such as "virtual organizations" and "science gateways". In this paper we explore the design options for attribute-based authorization in grid that will better address authorization and access control in such virtualized environments.

2. Virtual organizations and service oriented architectures

A virtual organization (VO) can be defined as a community of individuals that transcends physical organizational boundaries. For example, many computational projects conducted in grid infrastructures span multiple institutions. Individuals involved in such projects naturally form a virtual organization. The administration of such a VO can sometimes be delegated to one of the physical organization with which the members of the VO are affiliated. However sometimes it is not feasible to designate any one of the physical institutions as having the administrative rights over the VO; or the members of the VO do not want to deal with the physical organization's administration to help them maintain the VO membership information. In such cases, the VO members may choose to manage their membership by themselves. The implication of this approach on grid security is that information needed for authorization decisions may not always be available within the administration of a particular physical organization. For example, in typical settings, to determine whether a user has certain rights to use some resource, the resource provider can query a user database maintained by the user's home institution (e.g. an LDAP server) to retrieve the user's relevant information. However, if VOs are in place, it may not always be possible to retrieve such information from a user's home institution, because information about VO membership may not be maintained there. Authorization policies must thus be flexible enough to support the specification of different trust relationships: for certain categories of user information the resource provider trusts the user's home institution; but for user's membership regarding a VO, the resource provider may trust a database maintained by the VO's members.

In parallel, recent years have seen a shift in grid computing towards the adoption of *service oriented architectures* (SOA), in which a user does not directly interact with grid infrastructures but rather accesses the grid through a *service provider* (SP). The SP in turn makes requests to the resource provider (grid site) on the user's behalf. There are several reasons why this architecture has become popular, the most important one being that the SP can maintain a collection of applications of particular interest to a user community. The users of the SP can directly use those applications without having to obtain the application code and uploading them to the grid sites. We distinguish between SP and resource providers (RP's) even though it is understood that RP's which actually operate and maintain the hardware resources necessary to execute services could also be SP's. We intentionally separate

the SP's and RP's to tackle the case when a scientific community may build its own infrastructure and offer tailored services to its user community. These services in turn may use other services offered by the RP such as job execution, file transfer and so on. Indeed, there is a trend to move service offerings one level up in the hands of the actual scientific communities which best know their computing needs are and how they want to interact with the services to enable more science. This trend is illustrated by the TeraGrid science gateway program [9] whose aim is to outreach to communities as a whole and to support a generic access to TeraGrid resources so that a large number of scientists in the nation may benefit from the TeraGrid resources. In some sense the grid security challenges resulting from the addition of VO and science gateways are related. Both require the authorization system to handle large groups of users. There may be thousands of users in a VO and there may be thousands of users accessing grids through a SP. This requires the back-end grid infrastructures to have the ability to authorize a community of users in an effective and scalable way. However, current authorization mechanisms in place in grid systems do not properly address this requirement yet, as we illustrate in the next section.

3. Approaches to authorization for grid systems

Unlike in a single-host UNIX system, in a grid it is not possible to allow every authenticated user to use the protected resources. The authentication process just proves that the request is from a user, for example John Smith. Whether John Smith can use a computational resource in a particular grid site is another matter. A grid is inherently a federated environment, in which every local site wants to retain its authority on determining who can use its resources. Thus a separate authorization process is required to perform access control. In the Globus toolkit's GSI component, authorization is verified by consulting a grid-mapfile, which contains a mapping from a DN, a globally unique name assigned to a grid user, to a local account. Only a DN that has an entry on a grid site's grid-mapfile can use the site's computational resource. Once the user is authenticated, it is assigned to a local account according to the grid mapfile and can then run a job under this account. Further accesses are controlled to the local resources are controlled by the local system against such account. The use of an authorization mechanism based on the grid-mapfile is not scalable and it is not able to handle dynamic user populations. In architectures characterized by VO's and

science gateways, the number of users associated with a VO or SP may be huge, and moreover the membership of users in a VO or SP may change dynamically; it is not realistic to require the RP to keep track of these membership changes.

Currently, a number of projects are on-going to address such problem [4, 10] and they all adopt the notion of *attribute-based authorization*. In real life, authorization decisions are typically based on a user's role in an organization, rather than his unique identity. For example, a policy regarding the usage of a computer may contain a statement like "every faculty member of the university can use the machine", instead of listing the names of every faculty member. "Faculty member" is an attribute associated with a user describing the user's role in an organization. Policies written in terms of attributes rather than individual user's identity more accurately capture the high-level access control policies. Certainly, an authority needs to provide the attributes for every user that may need to use the resources. Such attributes should be provided by an attribute server trusted by the RP (for example, a server maintained by the university's human resources department could provide attributes regarding faculty membership). There are a number of dimensions to choose when building an attribute-based authorization system. Different options result in different attribute collection processes. We discuss the most relevant dimensions in what follows.

Push mode vs. pull mode. The *push strategy* requires the user to contact an attribute authority service to obtain attribute certificates and to "push" them to the target service when submitting a request. This approach allows the user to select the specific roles under which he would like to be authorized. The *pull strategy*, on the other hand, does not require the user to submit any attribute. The attributes are directly retrieved by the RP on behalf of the user.

IdP for institution vs. IdP for VO. In an attribute-based authorization system, an attribute server is also referred to as an Identity Provider (IdP). Some IdP's are associated with a physical organization such as a university. IdP's may also be associated with a VO. An important question in this context is how a user's attributes should be maintained given the diversity of attributes. Usually a user is associated with a "home institution", typically his employer. The home institution may seem to be the only entity entitled to provide a user's attributes. However, it is often the case that a user's attributes may not be related to his home

institution. A well-known example is represented the “IEEE” problem [19]. Also, it is very common for VO’s to intersect with multiple physical organizations and it is not always possible or desirable to assign one of those organizations the responsibility of maintaining the user’s attributes related to the VO. Thus, it is necessary to support flexible attribute maintenance schemes and rely on the use of multiple interoperating IdP’s.

4. Current projects investigating attribute-based grid authorization

Currently a number of projects are on-going to implement different attribute-based authorization schemes for grids. We now briefly describe the two most relevant ones: the VO Privilege Project [10] and GridShib [4].

The **VO Privilege Project** [10] has been developed by US CMS and US ATLAS in order to implement fine grained authorization for access to grid-enabled resources and services. The goal of the project is to improve user account assignment and management at grid sites. The Privilege Project implements the push authorization mode, discussed in the previous section; a grid user is asked to submit his identity attributes. The principle of least privilege access is enforced, to prevent accidental over usage of resources from authorized users and limit the damage a malicious entity can cause when a user's credentials are compromised. The implemented access control mechanism is illustrated in Figure 1¹.

VOMS is the server that provides users' attributes about VO membership. A VO first compiles a list of users that can use data. A user, before accessing the grid service first obtains a “token” (i.e., an attribute certificate) from the VOMS and embeds it into his grid proxy certificate. The user then presents the proxy certificate to the site when submitting a job or initiating a file transfer.

Operatively, when the proxy is forwarded to a gatekeeper, instead of consulting the gridmap file, the gatekeeper contacts the site-wide GUMS [5] service to map the request to a local account. The PRIMA module retrieves the attribute information from the user's proxy certificate and communicates it to GUMS. As a last step, the gatekeeper contacts the Site

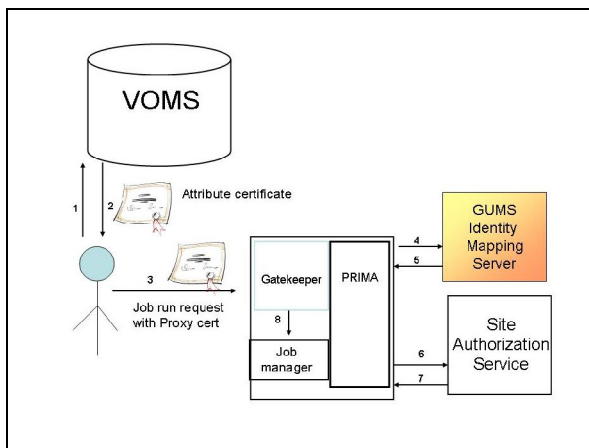


Figure 1: VO Privilege Project.

Authorization Service to enforce the site-specific access control policies.

GridShib [4] is an example of a pull mode authorization system. The GridShib authorization mechanism integrates Shibboleth [8] with the Globus Toolkit [2] and the Globus Secure Infrastructure (GSI). Shibboleth is an infrastructure for cross-identity authentication, which exploits the concept of federated identity information to federate user identity attributes. In Shibboleth, when a user at one institution tries to use a resource at another, Shibboleth sends attributes about the user to the remote institution, rather than making the user log into that institution. The receiver can check whether the attributes satisfy the SP's policy. The IdP in the Shibboleth architecture has all the user attributes and user privacy preferences which are taken into account when this IdP gives information to other SP's. In its current version, GridShib is implemented as a plug-in for Globus Toolkit 4.0 (GT4). The plug-in implements a policy decision point based on attributes obtained from a Shibboleth attribute authority. In a nutshell, the idea by GridShib is to authenticate grid users using GSI, determining the address of the Shibboleth attribute server in the process, and then obtain from the Shibboleth service the selected user attributes that the Grid service is authorized to access. Unlike other approaches, in GridShib the clients of services are not directly affected and do not even need to know that Shibboleth is involved in their decision-making.

¹ This is a simplified version of the diagram at <http://computing.fnal.gov/docs/products/voprivilege/>

Shibboleth in its original version, exploits handle-based authentication. GridShib instead replaces this approach with X.509 distinguished names to identify principal, thus achieving a better integration with existing PKI infrastructure. The service provider receives a proxy certificate in place of the handle typically issued by the IdP. GridShib is still under development and a number of open issues still need to be addressed. For instance, full integration between Shibboleth/SAML and Grid Security/X.509 is still to be achieved.

As described, the goals of the projects are very similar as they both aim at developing a flexible and efficient attribute based authorization mechanism for grid systems. The approach to the problem is different mainly with respect to the strategies used to collect users' attributes. User authentication is, in both cases, based on X.509 certificates, and local identities are mapped using the DN stored in the certificate. The possibility for a user to push his attributes as in the Privilege Project let him to select the role he wants to use for executing the request, though it requires additional operations at the user side. The pull mode adopted by GridShib is, however, easier to deploy, since the service clients are not affected and can submit jobs regardless the underlying authorization system used. Also, trust in GridShib is based on a bilateral arrangement between IdP and grid SP, by exchanging and consuming each other's metadata. In the Privilege Project, instead, trust needs to be established between the SP and the VO. Hence, in both cases, this approach results in a high number of trust relationships, which do not scale in grid systems composed by several entities.

It is not possible to establish which one better addresses the access control requirement we have outlined, as both the projects are still in progress. In particular, an evaluation of GridShib is premature, as it is at a preliminary stage. However, it is expected to be tested on a real case scenario within the TeraGrid [9] in the near future. Once the test bed will be completed, a more clear analysis may be performed. It is also worthy to note that GridShib in its final version will include both push and pull modes.

5. Approaches to authorization for Grid systems

Currently, command line access is the most common operation mode for grid applications. A user first authenticates to the grid site using his X.509 certificate. The site then makes its access control decisions either based on the grid-mapfile or the user's attributes and its

local policy. When accessing grids through a science gateway, a user does not necessarily authenticate directly to the grid site. Very often the authentication is between a user and a SP, which uses one or more existing grid infrastructures as its computational backend. This architecture poses a challenge to authorization: without the user's identity, how can the grid site know who is using its resources and whether to grant access or not? We have devised three possible authorization modes for such a service-oriented architecture. The different approaches are all characterized by three main entities: a user, a SP (the science gateway), and a RP (the grid site). Since the user does not interact directly with the RP, the RP has to rely on the SP to perform some of the authorization tasks. The difference among the three modes largely arises from the different levels of trust between the SP and the RP.

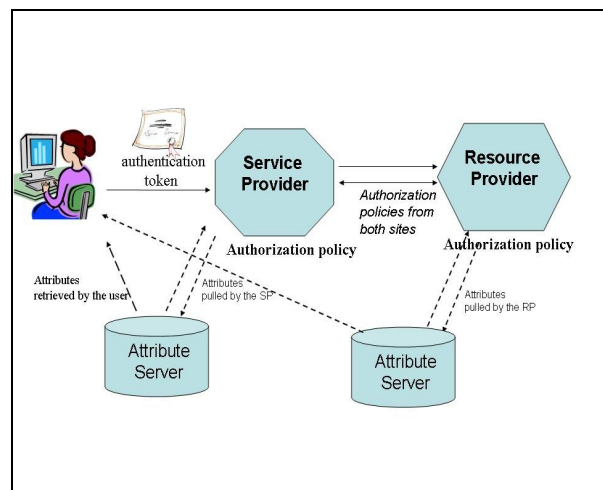


Figure 2: Service oriented architecture- medium trust.

Complete Trust. In this case, the SP is trusted completely by the RP. The SP enforces both authorization and authentication, while the RP is not involved at all in such process. The SP thus acts as a VO that provides both authentication and authorization services. The authorization is based on attributes, collected from an attribute server. Note that, although the attribute server and the service provider are two separate entities, they could be managed at the same site. The user is essentially anonymous at the RP side — the only information the RP gathers from the request is related to the specific job to execute. The SP negotiates resource allocations with the grid site and its users do not need to be aware that they are using the grid as the computational backend.

Medium Trust. In this case, the SP still performs authentication, but the final authorization decision is taken by the RP. An identity token is passed from the SP to the RP that does not necessarily contain the user's unique identity, but rather a user's attributes or a handle that can be used later to retrieve a user's attributes. These attributes should be provided either by the SP or another trusted third party. Since the authentication is between a user and the SP, the name space of the users' identity does not necessarily apply to the RP. For example, the grid site would not recognize a user John Smith who is registered only with the SP². Generally speaking, the user's identity cannot be directly used by the RP to reach an access control decision. Rather, the RP can base its authorization decisions on the user's attributes. These attributes should be provided by the SP or the federation the SP belongs to. For example, an SP may provide attributes reflecting a user's paid premium and request the RP to schedule jobs on different priorities based on the premiums. Or the SP can provide attributes reflecting a user's trustworthiness --- an anonymous user is less trustworthy than a registered user. The RP can then apply different access control policies based on a user's trustworthiness. The advantage of this approach, compared with the first one, is that the RP can differentiate its service based on the attributes provided by the SP. The RP still trusts SP to a considerable degree because the authorization decision is based completely on the attribute information provided by the SP. The RP can however maintain its own control of authorization policies instead of delegating such job to the SP. Additionally, under this mode, the SP can enforce its own authorization policies, if needed. A request may be first filtered by the SP's policy before even reaching the RP. Such specific case is shown in Figure 2, in which an arrow connects SP and RP and both enforce their own authorization policies and both SP and RP have access to an attribute server each.

A possible approach for implementing this authorization mode is to employ the SP as a weak certification authority (CA) for the RP. A weak CA is a certificate authority entitled to issue short-term user credentials for authentication purposes. Such certificate could even contain a meaningless DN, and would however allow the RP to recognize different users. This

² We do not exclude the possibility that the SP and RP share the same name space for user identities, but we do not restrict our analysis on this assumption.

type of short-term certificate is also called "junk certificate", because the DN's on those certificates do not represent user names that are meaningful for the grid site. Such DN's cannot be directly used in any authorization decision, but they can be used to retrieve user's attributes from the SP. Both the RP and SP can make authorization decisions based on users' attributes. Attributes might be retrieved using either the push strategy or the pull strategy (see Section 3). The attribute authority may be the SP itself, or some other third party trusted by the SP and/or the RP. Shibboleth [8] can play an important role in this model. The junk DN can serve as the Shibboleth handle. The SP takes the role of the IdP in Shibboleth, which executes authentication and issues handles for users. The attribute servers, though logically separate from the SP, could be the same entity in practice. Otherwise the mapping between the handle and user attributes must be propagated from the service provider to the attribute server whenever a new user is authenticated.

No trust. In this case, both authentication and authorization are executed by the RP. A user who already has a resource allocation at RP can access the grid through the SP using his own grid credentials. Operatively, to implement this approach the SP may be informed of the content of the gridmap file at the RP and thus check if the user belongs to the list. Then, a secure channel between the RP and the users, passing through the SP can be created. It is thus a task of the SP to authenticate the requester and evaluate whether or not it possesses sufficient credentials for being authorized.

Note that hybrid solutions are also possible. A possible approach to support both the second and the third scenarios may be that of a SP keeping track of users that already have a resource allocation. In this case, when a job request is submitted, the SP can check if the DN the user presents for authentication belongs to such list. If it does, then a secure channel is created. If the DN is not part of the list of known DN's, the SP will apply its own authorization policies, which may require retrieving the user's attributes.

5. Concluding remarks

In this paper we have discussed the need for attribute-based authorization technology in grid computing to accommodate the emerging trend of virtualized environments, under which users access grid resources as a virtual community through a service

provider. We have discussed several design dimensions of an attribute-based authorization system for grids and discussed their advantages and disadvantages with regard to this trend. We have also outlined solutions for attribute-based authorization for accessing grids through science gateways, and discussed how existing/ongoing grid authorization projects in this area could be leveraged to build such systems.

10. References

- [1] Condor, High Throughput Computing. <http://www.cs.wisc.edu/condor>.
- [2] Globus Toolkit. <http://www.Globus.org>.
- [3] Grid portal use case for GridShib. <https://authdev.it.ohiostate.edu/twiki/bin/view/GridShib/GridPortalUser>.
- [4] The GridShib project. <http://gridshib.globus.org/>.
- [5] The GridShib project. <http://grid.racf.bnl.gov/GUMS/>.
- [6] NanoHUB. <http://www.nanohub.org>.
- [7] Overview of the grid security infrastructure. <http://www.globus.org/security/overview.html>.
- [8] The Shibboleth. <http://shibboleth.internet2.edu/>.
- [9] The TeraGrid project. <http://www.teragrid.org>.
- [10] The VO Privilege Project. <http://computing.fnal.gov/docs/products/voprivilege/>.
- [11] T. Barton, J. Basney, T. Freeman, T. Scavo, F. Siebenlist, V. Welch, R. Ananthakrishnan, B. Baker, and K. Keahey. Identity federation and attribute-based authorization through the Globus toolkit, Shibboleth, GridShib, and Myproxy. In *5th Annual PKI R&D Workshop*, October 2005.
- [12] J. Basney, M. Humphrey, and V. Welch. The MyProxy Online Credential Repository. In *Software: Practice and Experience*, volume 35(8), 2005.
- [13] D. Chadwick. Authorization in Grid Computing. *Information Security Technical Report*, 10(1):33–40, 2005.
- [14] J. Fortes, R. Figueiredo, and M. Lundstrom. Virtual computing infrastructures for nanoelectronics simulation. *Proceedings of the IEEE*, 93(10), August 2005.
- [15] I. Foster and C. Kesselman, editors. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [16] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150, 2001.
- [17] OASIS. Xacml 2.0 approved as oasis standard.
- [18] W. F. R. Housley, W. Polk and D. Solo. Internet X.509 Public Key Certificate and certificate revocation list (CRL). RFC, 3280, network working group, April 2002.
- [19] V. Welch, T. Barton, K. Keahey, and F. Siebenlist. Attributes, anonymity, and access: Shibboleth and globus integration to facilitate grid collaboration. In *4th Annual PKI R&D Workshop*, April 2005.