Assured resource sharing in Grid environments

Jing Jin

Deutsche Bank Global Technology,
3000 Centre Greenway,
Cary, NC 27617, USA
E-mail: jjin@uncc.edu

Gail-Joon Ahn*

Laboratory of Security Engineering for
Future Computing (SEFCOM),
Arizona State University,
P.O. Box 878809,
Tempe, AZ 85287, USA
E-mail: gahn@asu.edu
*Corresponding author

Abstract: In Grid-based collaborations, a number of data sharing services in Grid are established to provide a unified platform for dynamic discovery, access and replication of distributed data. Controlling access to Grid data in these services requires the ability to dynamically make authorisation decisions based on the data owners' policies and users' credentials across administrative domains. In this paper, we present a flexible policy-driven authorisation system, called RamarsAuthZ, for secure data sharing services in Grid systems. RamarsAuthZ adopts a flexible role-based approach with trust-aware feature to advocate originator control, delegation and dissemination control. A case study based on Globus data replication service (DRS) is presented to provide effective access control both at the service level and at the data level. Our system is flexible and interoperable with multiple Grid services with little reliance on static policy and attribute management.

Keywords: access control; Grid systems; assured sharing; security.

Reference to this paper should be made as follows: Jin, J. and Ahn, G-J. (2011) 'Assured resource sharing in Grid environments', *Int. J. Information and Computer Security*, Vol. 4, No. 3, pp.215–233.

Biographical notes: Jing Jin received her PhD from the College of Computing and Informatics, University of North Carolina at Charlotte, Charlotte. She was a member of the Laboratory of Information Integration, Security, and Privacy (LIISP), University of North Carolina at Charlotte. Her current research interests include access control and trust management, identity and privacy management, network and distributed system security and security in health informatics.

Gail-Joon Ahn is an Associate Professor in the School of Computing, Informatics, and Decision Systems Engineering and the Director of SEFCOM at the Arizona State University. His research interests include information and systems security, vulnerability and risk management, and security architecture for distributed systems. His research has been supported by the US National

Science Foundation, US Department of Defense, US Department of Energy, Robert Wood Johnson Foundation, and Bank of America. He is a recipient of the US Department of Energy CAREER Award. He received his PhD in Information Technology from the George Mason University, Fairfax, Virginia in 2000.

1 Introduction

Grid computing represents an important infrastructure for coordinated pooling and sharing of resources transcending institutional boundaries. Common functionalities in Grid applications are to manage, provide access to, and integrate large volume of data at one or multiple sites. Grid data and resources by nature are diverse in their locations, types, structures, ownerships, naming conventions and access capabilities. By embracing the service-oriented approach, such complexities have been made transparent to clients through a layer of virtualised Grid data sharing service providing a unified platform for data discover, access and replication (Foster et al., 2002; Raman et al., 2002). Examples of such Grid data sharing services are Globus data replication service (DRS), open Grid services architecture – data access and integration (OGSA-DAI) and storage resource broker (SRB) (Foster, 2006; Antonioletti et al., 2005; Rajasekar et al., 2002). However, the facility provided to Grid clients requires more advanced Grid access control mechanisms to accommodate various challenges ranging from the authorisation model to the system architecture and deployment.

Firstly, Grid systems are usually composed of a number of dynamic and autonomous domains involving a large number of distributed users, and different domains have their own security policies. Attribute-based access control, which makes decisions relying on attributes of requesters, resources, and environment, has been widely adopted as a scalable and flexible authorisation solution for Grid environments (Lang et al., 2008). In an attribute-based authorisation system, the entity that manages user attributes is referred to as an identity provider (IdP). A user's attributes are normally collected by multiple IdPs in Grid systems. For example, a user is associated with a 'home institution' which typically manages his employment status and affiliation attributes, while another IdP is associated with a Grid virtual organisation (VO) that maintains attributes such as membership and role information. Therefore, authorisation systems need to support scalable and flexible attribute-based access control to deliver users' attributes from multiple IdPs in a secure and trusted manner.

Secondly, as various data resources are shared through the Grid data sharing service, the data owners should participate directly in defining authorisation policies for their data sets, and their authorisations need to be efficiently conveyed and enforced within the Grid data sharing service. In addition, the Grid data sharing service itself apparently is a type of Grid resource where its access and invocation needs to be well protected according to the resource provider's (RP's) security policies. Therefore, it is required for authorisation systems to be flexible enough to synthesise both service-level and data-level controls accommodating security policies from different stakeholders such as the data RPs and the service providers (SPs).

Finally, from the system architecture and deployment perspective, there are a number of dimensions to be considered for an attribute-based authorisation system. In terms

of attribute collection process, the 'push' strategy requires the clients to provide the attributes ahead, obtaining and pushing those attributes to the Grid service at the initial request. The 'pull' strategy, on the other hand, does not require the clients to submit any attribute. Instead, it is the responsibility for a Grid authorisation system to acquire attributes from the clients' IdPs. While the clients have more options to select the attributes being released for authorisation in the 'push' mode, the 'pull' mode simplifies the overall interception by the clients. It is impossible to determine which mode is more suitable for dynamic Grid environments. However, it is highly desirable for authorisation systems to be flexible enough to cope with both options. In terms of system deployment, there are usually a number of Grid services running within Globus Toolkit. To make an authorisation system more flexible to serve the authorisation functions for these Grid services, the reliance on statically configured modules to render an authorisation decision such as policy and attribute management needs to be minimised.

There are continuous attempts to develop a common attribute-based authorisation framework for Grid systems, such as Shibboleth, VOMS, Akenti, and Permis (Cantor, 2005; Alfieri et al., 2003; Thompson et al., 1999; Chadwick et al., 2006). However, these systems suffer various limitations in accommodating the above-mentioned requirements. In this paper, we present a flexible policy-driven authorisation system, called RamarsAuthZ, for secure data sharing services in Grid systems. RamarsAuthZ adopts a flexible role-based approach with trust-aware feature to advocate originator control, delegation and dissemination control. A case study based on Globus DRS service is presented to provide effective access control both at the service-level and at the data level. The rest of this paper is organised as follows. In Section 2, we give a brief overview of existing attribute-based access control systems in Grid computing. In Section 3, we introduce our proposed authorisation framework and discuss the integrated RamarsAuthZ system design. Section 4 describes the performance evaluation and followed by a brief comparison between RamarsAuthZ system with existing work. We conclude our paper with future research directions in Section 5.

2 Related work

There are a number of attribute-based access control systems proposed for Grid environments. Shibboleth (Cantor, 2005) is an attribute authority service developed by the Internet2 community for cross-organisation identity federation. A Shibboleth IdP asserts attributes about a user as SAML assertions, and the relying parties can make access decisions based on these attributes. In VOMS (Alfieri et al., 2003), every Grid VO manages its own members, and a Grid user can access the available resources by obtaining and 'pushing' an X.509 attribute certificate (Housley et al., 2002) containing his VO membership to the resource. In Groeper et al. (2007), the authors proposed an approach where Grid users are able to collect attributes both from VOMS and Shibboleth, so that authorisation can be made based on the attributes from both sources. The Akenti system (Thompson et al., 1999) represents the authorisation policies for a resource as a set of certificates digitally signed by multiple distributed stakeholders. These certificates express the attributes a user must have in order to get specific rights to a resource. Akenti allows the certificates to be stored in distributed remote repositories and provides mechanisms based on the 'pull' architecture to ensure that all applicable usage conditions are combined when making an access control decision.

Permis leverages the role-based access control utilising X.509 attribute certificates, and it has been recently integrated with Shibboleth to retrieve the role attribute of a user (Chadwick et al., 2006). Also, Laccetti and Schmid (2007) introduce a unified approach for access control to Grid resources based on PKI and PMI infrastructures at the Grid layer, ensuring that an adequate transfer of authentication and authorisation is made between the VO and RP layers.

These authorisation systems, however, rely on static configurations of their own policies and attribute providers (Aps), and cannot support dynamic policy and attribute discoveries for authorisation. A more flexible and scalable attribute-based access control method is still needed to achieve more effective access control for heterogeneous Grid environments.

2.1 Globus toolkit

The open source Globus toolkit (Globus Alliance, 2006) is the core Grid infrastructure that provides all necessary functionalities for running Grid jobs including resource monitoring, discovery, and management. Inside Globus toolkit version 4 (GT4) which is the latest release of the toolkit, the Grid security infrastructure (GSI) (Welch et al., 2003) is implemented as the de-facto solution to provide the fundamental security services such as authentication and message protection for Grid environments. In particular, X.509 proxy certificate (Tuecke et al., 2004) is utilised in GSI to allow a Grid user to periodically delegate his identity and privileges to another entity so that the bearer is able to authenticate and establish secure connections with other parties on the Grid user's behalf. In our approach, we explore the proxy certificate to convey attributes and other necessary information from Grid client to RamarsAuthZ authorisation system so that the system can be automatically configured for the 'push' or 'pull' modes.

GT4 server-side authorisation framework encapsulates a set of built-in policy decision point (PDP) modules. The default authorisation PDP in GT4 evaluates an access control list (ACL) type of policy located in *Grid-mapfile*, which specifies mappings of a user's global identity (called distinguished name, DN) to a local account. Users are authorised to use the resources when their DNs appear in such list and the privileges are determined by the associated local account. This authorisation approach is primitive and does not scale to a large number of Grid users. More recently, the global Grid forum (GGF) has proposed a SAML AuthZ specification (Welch et al., 2006) using SAML (OASIS, 2003) as a standard message format for requesting and expressing authorisation assertions so that external authorisation systems can remotely make authorisation decisions and respond authorisation queries. With this approach, it is required for the external authorisation system to render an access decision based on the information conveyed by SAML request with the least dependency of static configurations. Our RamarsAuthZ authorisation system fully supports the SAML-based method to make authorisation decisions for the enhanced DRS.

2.2 Globus DRS

GT4 provides a number of components to enable collaborative data sharing concerning with the discovery, access and dissemination (Foster, 2006). The replica location service (RLS) is responsible for the data registration and enables the discovery of data resources.

Inside an RLS, a unique identifier called *logical name* is created for each registered data item. A mapping is maintained from the *logical name* to the *physical locations* of the data item and its replicas. For instance, 'GeneSequence – gsiftp://abc.com/var/gseq.tar' states an entry in RLS. By querying the logical name 'GeneSequence', a user could locate his desired data item at 'gsiftp://abc.com/var/gseq.tar'. When a data resource is discovered, the reliable file transfer service (RFT) is in charge of data replication to the target location. Based on RLS and RFT, Globus DRS is developed as a higher-level data management service incorporating the functionalities of both RLS and RFT services. In particular, upon receiving a user's file replication request, the DRS begins by querying RLS to discover the existence of the desired files. Then the DRS invokes RFT to replicate the files. When the file transfers are completed, the new replicas are finally registered with RLS so that they can be further discovered and disseminated. In terms of authorisation, the DRS service alone can be configured with GT4 built-in authorisation mechanisms. Therefore, only authorised Grid users can invoke the service and exercise the data replication functions. However, such configuration cannot further protect the actual data resources that are replicated through DRS. Our system largely enhances the authorisation in DRS by enforcing both the service-level and data-level controls based on different stakeholders' policies.

2.3 Shibboleth and GridShib

Shibboleth is an Internet2 middleware initiative project aiming at providing cross-domain single sign-on and attribute-based authorisation based on SAML (Cantor, 2005). Shibboleth leverages the identity federation between educational organisations so that a user can authenticate on his own campus and access to remote resources where his attributes are passed to the RPs for authorisation. There are two major separately deployed components in Shibboleth infrastructure: the identity provider (Shib-IdP) and service provider (Shib-SP). A Shib-IdP manages a user's identities and asserts his attributes. A Shib-SP, on the other hand, resides at the resource side to request the remote user's attributes from his Shib-IdP and enforces attribute-based access control. All participating Shib-IdPs and Shib-SPs are managed in a Shibboleth federation, where their trust relationships are bridged through the centralised Shibboleth federation certificate authority.

GridShib (GribShib Project, 2008) project is initiated to integrate Shibboleth infrastructure with Grid technology. A set of tools has been developed to facilitate Grid resources to fetch attributes from Shibboleht IdP. However, very limited attribute-based authorisation functionalities can be achieved by GridShib. Most of early adopters (Chadwick et al., 2006; Groeper et al., 2007) of GridShib only implement the 'pull' mode of attribute acquisition, relying on pre-configured single source of Shib-IdP. This approach lacks flexibility and is insufficient to support the attribute-based access control based on multiple IdP sources. In addition, as Shibboleth federation is separated from the common trust base established in Grid VO, the attributes issued by Shibboleth IdP are not necessarily trusted by Grid resources. The trust management issue is not well addressed either in current available systems. In our research, we explore a feasible and integrated solution to these issues.

3 Assured resource sharing in Grid environments

Role-based access management for ad-hoc resource sharing framework (RAMARS) has been proposed as a policy-driven role-based access management solution for the resource sharing in the ad-hoc collaboration environment (Jin and Ahn, 2006; Jin et al., 2007). RAMARS advocates the originator control where the owner of the data resource, also called an originator, has the ultimate authority over its data resource to clearly define who is authorised to access the data and to what extent the data information can be distributed. In particular, an originator does not rely on any established security services to maintain membership and privileges. Instead, the originator maintains its discretionary sharing control domain by defining a collection of collaborator roles. And specific data sharing capabilities for data discovery, data access and data dissemination are delegated to these collaborator roles. Remote users are dynamically included in the originator's sharing control domain to gain access privileges by being assigned to these roles. Unlike the traditional RBAC that the users are identified and assigned based on their identities, RAMARS introduces another layer of abstraction, where users are assigned to roles based on their attributes. An originator defines a set of attributes that a user must possess for the user to be assigned to a particular collaborator role, through which an originator could easily manage the delegated sharing capabilities. Remote users should present credentials to claim the possession of required attributes, yet it is up to the originator's discretion to validate and determine the trustworthiness of these credentials, thereafter to decide whether the claimants of certain attributes can be accepted for the role assignment. In RAMARS, the determination of trustworthiness of user attributes is handled through a special type of trust management constraint, where the delegation of authority is considered as an important mechanism for an originator to manage the degree of trust with different attribute authorities. The scheme can be easily adapted into the Grid environment by delegating the authority to trusted Grid VOs and/or Shibboleth IdPs.

As a policy-driven approach, all the salient features of RAMARS are realised in a collection of policy components using standard XACML policy language (OASIS, 2005). The policy set consists of the following components.

- root meta policy set (RMPS) is the top-level policy for an originator to declare the ownership of the data resource and specify the location of the originator's ROA policy set. This enables the distributed policy deployment in RAMARS system architecture, and the authorisation system could dynamically locate and retrieve the ROA policies
- role-based originator authorisation policy set (ROA) maintains the core authorisation and trust management policies for an originator to govern its control domain and delegate data sharing capabilities. The policy set contains the following subpolicies:
  1. role policy set (RPS) defines a set of collaborator roles within an originator's control domain
  2. capability policy set (CPS) specifies the sharing capabilities assigned to each collaborator role

3 role assignment policy set (RAPS) defines the required attributes for a remote user to be assigned to a certain collaborator role

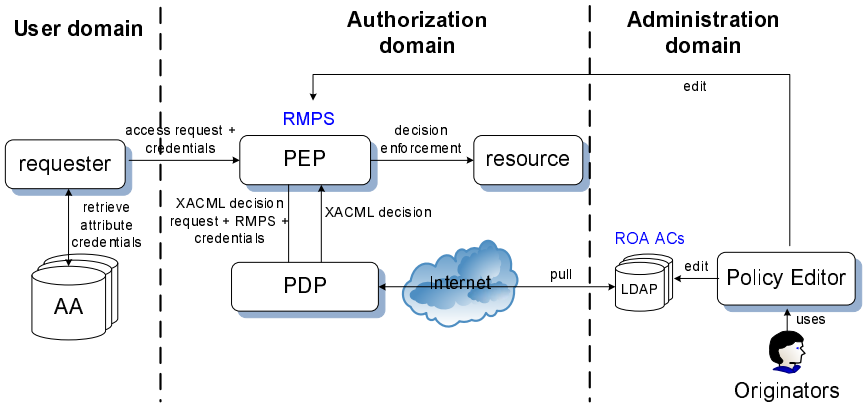
4 trust assessment policy (TAP) defines two internal policies to evaluate the trustworthiness of a user's attributes. A trust level assessment policy specifies the rules to determine the trust level of a user's claimed attributes given the supportive credentials. And a trust decision policy is specified to determine the trustworthiness of the claimed attributes given the trust level. Only trusted attributes can be promoted for role assignment

- credential policies (CRED) are specified by attribute authorities to assert user attributes and associate validation rules (e.g., validity period) to the credential.

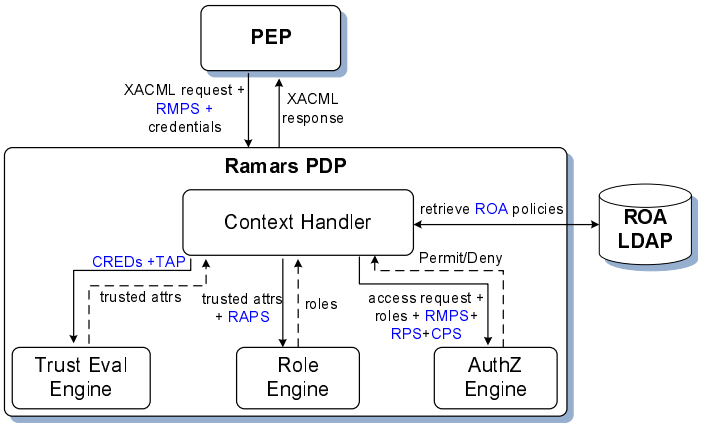
RAMARS system is designed to be deployed in distributed collaborative environments without assuming any centralised policy store. The architecture of RAMARS system can be segregated into three domains. In the administration domain, an originator edits and maintains ROA policies in its administrative domain. These policies are expected to be automatically enforced by the authorisation system without further interception by the originator. In the authorisation domain, both the policy enforcement point (PEP) and RAMARS PDP are supposed to be integrated with collaborative sharing applications to protect the data resource by enforcing the originator's ROA policies. The root RMPS policy is always associated with the data resource for RAMARS PDP to locate the originator's ROA policies. Upon receiving an access request and supportive credentials coming from the user domain, the PEP invokes RAMARS PDP with formulated access decision request, RMPS and the user's credentials. RAMARS PDP consists of four subcomponents: context handler, trust evaluation engine, role engine and authorisation engine. In particular, the context handler dynamically retrieves ROA policies from the originator's policy store based on the location references specified in RMPS. The requester's credentials and the originator's TAP policies are sent to trust evaluation engine where trusted attributes are derived. These trusted attributes and RAPS policy are carried by the role engine to determine the user's roles. And finally the authorisation engine makes the final access decision (e.g., *permit* or *deny*) based on the user's roles. This access decision is sent back to PEP as an XACML response for decision enforcement. Figure 3 illustrates RAMARS system architecture.

In the original RAMARS system, the user's credentials are pushed to the authorisation system, while the originator's ROA policies are dynamically pulled at runtime upon policy evaluation. Therefore, the deployment of RAMARS PDP does not need to be configured with any centralised policy and credential stores. In our RamarsAuthZ system, we enhance the existing system by supporting the hybrid 'push' and 'pull' modes for attribute acquisition. And we also extend the RAMARS PDP as a remote Grid authorisation service exchanging SAML authorisation messages with Grid services following the AuthZ SAML specification proposed by the open Grid forum (Welch et al., 2006).

Figure 1 RAMARS system architecture (a) RAMARS architecture (b) RAMARS PDP and policy evaluation (see online version for colours)



(a)



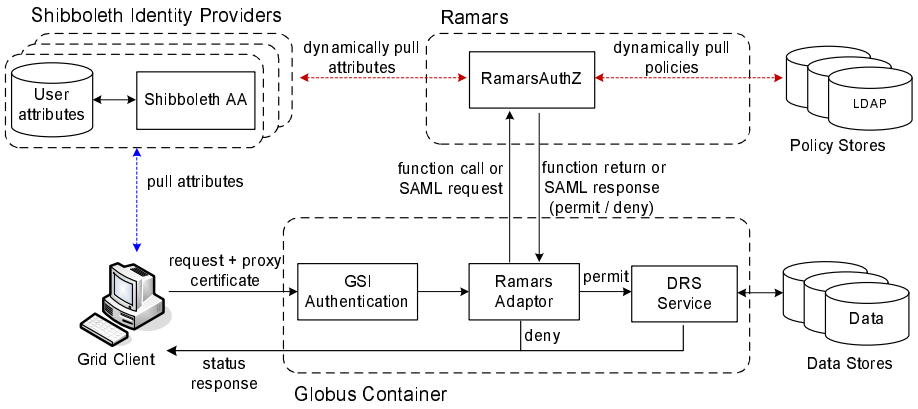
(b)

3.1 RamarsAuthZ: design and system architecture

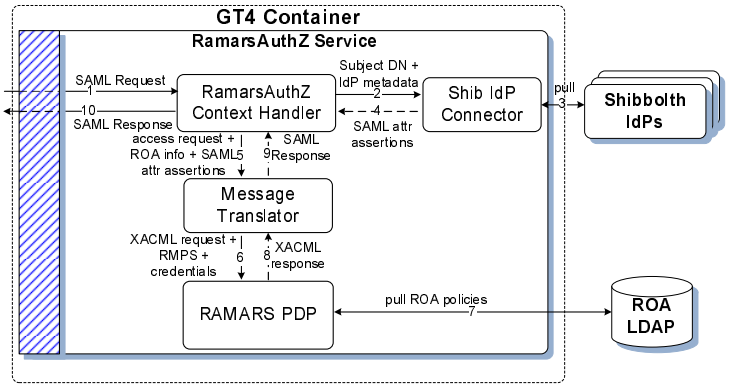
Figure 2(a) illustrates the system architecture for the integrated RamarsAuthZ system. Inside the Globus container, a RAMARS Adaptor is introduced for the Grid service (e.g., DRS service) to communicate with the RamarsAuthZ service which is essentially based on RAMARS PDP. The RamarsAuthZ service can be deployed and called out by the RAMARS Adaptor through two mechanisms: a localised function call by API or a remote service invocation by SAML messages. In terms of attribute acquisition, both 'push' and 'pull' modes are supported. A Grid client can retrieve his attribute assertions from various IdPs and 'push' these assertions to the Grid service in his proxy certificate (Tuecke et al., 2004). Alternatively, the Grid client can embed the metadata information about his preferred IdPs within his proxy certificate so that RamarsAuthZ is able to dynamically locate and retrieve the attributes for authorisation. The IdP metadata

contains important information for RamarsAuthZ to locate and communicate with the IdP, such as the IdP's unique providerId, the endpoint location, the IdP's certificate for SSL communication, and so on. We call this proxy certificate with specialised extensions as a RamarsAuthZ proxy certificate. A *Ramars-proxy-init* tool is introduced to facilitate a Grid user to create a RamarsAuthZ proxy certificate. By including these information in the legitimate extension fields of X.509 certificate, the RamarsAuthZ proxy certificate can be accepted and verified by the GSI authentication module in GT4 (Welch et al., 2003) without requiring any further changes. Since a proxy certificate is a self-issued certificate by the Grid user, all embedded attribute assertions and IdP's metadata must be signed by the issuing IdP and the metadata distributor, respectively. Any unsigned attribute assertions and IdP metadata without proper integrity protection are discarded and cannot be used by RamarsAuthZ for the authorisation decision.

Figure 2 Integrated RamarsAuthZ authorisation system (a) integrated RamarsAuthZ authorisation system overview (c) RamarsAuthZ service internal architecture (see online version for colours)



(a)



(b)

The overall flow of RamarsAuthZ authorisation and DRS service invocation works as follows: a Grid client sends a RamarsAuthZ proxy certificate and his data replication request to the Grid DRS service. Upon receiving the request, the client is first authenticated through GSI authentication module. Then RAMARS adaptor is invoked to parse the extensions in the proxy certificate and prepare an authorisation request for RamarsAuthZ service to check whether the Grid client is authorised to invoke the DRS service. The authorisation request includes information on the requester's attributes and/or preferred IdPs passed by the proxy certificate, and the location of the SP's ROA policies, which is specified in the service's security descriptor file when the service is deployed in GT4. Based on the authorisation request, the RamarsAuthZ service can dynamically retrieve the SP's ROA authorisation policies and the Grid client's attributes to make the authorisation decision. The decision is sent back to RAMARS adaptor and enforced accordingly by the DRS service.

The invocation of RamarsAuthZ is fairly simple via function calls when RamarsAuthZ service is deployed locally. Yet it is more challenging when RamarsAuthZ service is deployed remotely as a Grid authorisation service. In particular, the RamarsAuthZ service must be exposed through a standard SAML callout interface, and SAML is the only media for requesting and expressing authorisation assertions and decisions from RamarsAuthZ service. The SAML AuthZ specification (Welch et al., 2006) defines two SAML extensions for message exchanges between the calling service and the Grid authorisation service consisting of an *ExtendedAuthorisationDecisionQuery* (simply called *query*) flowing from the PEP to PDP, with a returned assertion containing a *SimpleAuthorisationDecisionStatement*. We further explore the *query* to include authorisation related information for RamarsAuthZ service. In particular, all user's pushed SAML attribute assertions are included as *evidence* in the *query*. And the ROA policy information and the user's IdPs information are encapsulated as the extended *AuthorisationAdvice* elements of *ROAReferenceAdvice* and *AAReferenceAdvice*, respectively. With a 'permit' authorisation decision sending back from the PDP, the Grid client's data replication request is executed by the DRS service. Figure 2(b) illustrates the internal architecture of RamarsAuthZ service. All communications to and from the service are through a standard SAML interface. Inside the service, the RamarsAuthZ context handler is responsible to parse the incoming SAML *request*. Then Shib IdP connector is invoked to initiate communications with the specified IdPs to acquire the user's attribute assertions. The message translator is responsible to convert the messages from SAML to the XACML-based input accepted by the original RAMARS PDP. With conflicting message formats being transformed, the policy evaluation is carried out by the original RAMARS PDP without any additional changes.

3.2 Enhanced DRS

As discussed in the Introduction, data resource originators delegate the data sharing responsibilities to the DRS service, the DRS service is then responsible to enforce the data originators' authorisation policies on their behalf during each step of data sharing

process, including the *data discovery*, *data access*, and *data dissemination*. We demonstrate such capabilities by implementing an enhanced DRS service.

In the original DRS service, the *data discovery* functionality is performed by a service called RLS registry, where a *logical name–physical location* mapping is maintained for each data resource and its replicas. For instance, 'GeneSequence – gsiftp://abc.com/var/gseq.tar' states an entry in RLS. By querying the logical name 'GeneSequence', a user could locate his desired data item at 'gsiftp://abc.com/var/gseq.tar'. After the physical location of the data item is successfully located, the RFT service component in DRS is invoked to copy the user's desired data items to the target locations. This realises the step of *data access*. And when the file transfers are completed, the new replicas are finally registered back in RLS so that they can be further discovered and disseminated, which illustrates the process of *data dissemination*. As we have demonstrated in Section 3.1, the DRS service can be protected by RamarsAuthZ so that only authorised Grid users can invoke the service and exercise the data replication functions. However, such configuration cannot further protect the actual data resources that are replicated through DRS.

In order to enable the originator control for each data resource being shared through DRS, we add additional attributes associated with each RLS entry to indicate the originator and its ROA policy information of the data resource as *originator* and *roa_location*, respectively. With these two attributes being specified, the DRS service not only can discover the physical location of the data resource, but also can collect the necessary ROA information for the RamarsAuthZ service to locate the data originator's policies. In the enhanced DRS, the access control for *data discovery* is conducted when DRS receives the response from the RLS registry. An access request is generated to query RamarsAuthZ service whether the requester is authorised to '*query*' the physical location of the requested data file. A 'deny' decision results in the failure of file location. Otherwise, before the RFT service is invoked for the file replication of successfully located files, the access control for *data access* has to be enforced, where the RamarsAuthZ service checks whether the requester is authorised to '*replicate*' the data, and a 'permit' decision can trigger the replication operation in RFT. After the data file is successfully replicated, a final *data dissemination* authorisation request is sent to RamarsAuthZ service to check whether the requester is authorised to further '*disseminate*' the replica to others. The replica is registered back in the RLS registry for other DRS queries only if the requester is authorised to do so.

Figure 3(a) shows a snapshot where a Grid client utilises the *Ramars-proxy-init* tool to generate the RamarsAuthZ proxy certificate before invoking the enhanced DRS service. The proxy certificate includes two SAML attribute assertions and metadata of two preferred IdPs as extensions. Figure 3(b) shows a snapshot for the RamarsAuthZ service, deployed as the 17th service in the GT4 container, to receive and process a SAML authorisation request.

Figure 3 RamarsAuthZ service (a) RamarsAuthZ proxy certificate generation (b) RAMARS PDP and policy evaluation

```
[jjin@coiti321 ramars_proxy]$ cat jjin.proxy.info
keyPath=etc/userkey.pem
certPath=etc/usercert.pem
samlAssertions=etc/assertion1.xml;etc/assertion2.xml
idpInfos=etc/idp1.metadata;etc/idp2.metadata
[jjin@coiti321 ramars_proxy]$ bin /ramars-proxy-init jjin.proxy.info
Enter GRID passphase:
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA-coit291.uncc.edu/OU=uncc.edu/CN=JJin
Creating proxy ................................ Done
Extension:  1.3.6.1.4.1.3536.1.1.1.32
Extension:  1.3.6.1.4.1.3536.1.1.1.32
[jjin@coiti321 ramars_proxy]$ /usr/local/globus-4.0.5/bin/globus-credential-dele
gate -h coiti321.uncc.edu -p 8443 mycredential.epr
EPR will be written to: mycredential.epr
Delegated credential EPR:
Address: https://152.15.98.177:8443/wsrf/services/DelegationService
Reference property[0]:
```

(a)

```
[15]: https://152.15.98.177:8443/wsrf/services/AuthzCalloutTestService
[16]: https://152.15.98.177:8443/wsrf/services/dai/DataResourceInformationService
[17]: https://152.15.98.177:8443/wsrf/services/RamarsAuthzService
[18]: https://152.15.98.177:8443/wsrf/services/WidgetNotificationService
[19]: https://152.15.98.177:8443/wsrf/services/dai/DataRequestExecutionService
[20]: https://152.15.98.177:8443/wsrf/services/dai/RequestManagementService
[21]: https://152.15.98.177:8443/wsrf/services/AdminService
[22]: https://152.15.98.177:8443/wsrf/services/ShutdownService
[23]: https://152.15.98.177:8443/wsrf/services/ContainerRegistryService
[24]: https://152.15.98.177:8443/wsrf/services/CounterService
[25]: https://152.15.98.177:8443/wsrf/services/TestService
[26]: https://152.15.98.177:8443/wsrf/services/TestAuthzService
[27]: https://152.15.98.177:8443/wsrf/services/SecurityTestService
[28]: https://152.15.98.177:8443/wsrf/services/ContainerRegistryEntryService
[29]: https://152.15.98.177:8443/wsrf/services/NotificationConsumerFactoryServices
[30]: https://152.15.98.177:8443/wsrf/services/dai/SessionManagementService
[31]: https://152.15.98.177:8443/wsrf/services/TestServiceRequest

Request is coming...
Subject: /O=Grid/OU=GlobusTest/OU=simpleCA-coit291.uncc.edu/OU=uncc.edu/CN=JJin

Retrieving ROA policies...
Retrieving attributes...
Authorization decision: Permit
```

(b)

3.3 Trust management

Trust management in our system is considered with two aspects: the organisational level concerning the necessary trust relationships between the involved parties and the technical level with respect to the implementation details.

At the organisational level, we consider four main entities, a Grid client, a Grid SP, the Grid RPs and the APs. In our research, the Globus DRS is a particular SP, the data resource originators are RPs and Shibboleth IdPs are APs. To simplify the analysis, we treat the RamarsAuthZ service as part of the SP as it provides authorisation decisions for the SP to enforce. Since the Grid client does not directly interact with the RP, the RP has to rely on the SP to perform some of the authentication and authorisation tasks.

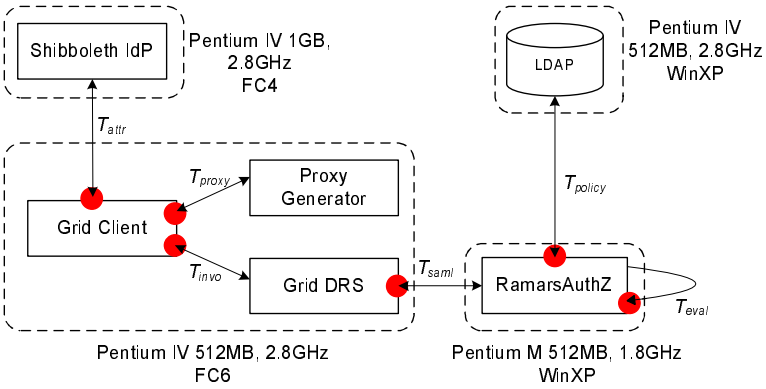
As a policy-driven approach, the RPs maintain their control through defining the ROA authorisation policies, within which the trust relationships between RPs and APs are explicitly defined as various trust levels in the trust assessment policies (TAP). However, the RPs still have to trust the SP to a considerable degree with that the authentication and ROA policies can be faithfully enforced. Additionally, the RPs also need to trust the APs for providing right attributes while these attributes are correctly handled by the SP associated with the RamarsAuthZ service.

At the technical level, the trust management among the Grid client, SP and RPs is carried out by the GSI infrastructure implemented in the Globus Toolkit where a Grid VO-CA is the centralised trust anchor. However, the APs leveraged by RamarsAuthZ for authorisation is managed in Shibboleth environment. It is necessary to bridge the trust relationships between these two distinct environments. In our implementation, we adopt the GridShib plugin (GribShib Project, 2008) to query Shibboleth IdPs (Shib-IdPs) for user attributes. The trust relationship is based on a bilateral arrangement between the two parties by exchanging and consuming each other's metadata. In other words, the RamarsAuthZ service maintains a set of certificates identifying trusted IdPs for authentication and secure communication purpose, while those IdPs keep the certificate of RamarsAuthZ. Therefore, with n entities being involved, there are $O(n^2)$ bilateral relationships to be managed. With the Shibboleth federation being involved, a single trusted Shibboleth federation CA is introduced to all involved Shib-IdPs and relying parties. This could partially ease the trust relationships managed by the Shib-IdPs and the RamarsAuthZ. In particular, it is possible for RamarsAuthZ to maintain two parallel certificates: one is issued by Grid VO-CA for authentication with all parties in Grid environments, and the other is issued by Shibboleth federation CA for the RamaraAuthZ to retrieve attributes from Shib-IdPs. In addition, other approaches such as bridge CA and online CA can be further explored for more seamless integration solutions on the trust relationships between Grid VO-CA and Shibboleth federation CA.

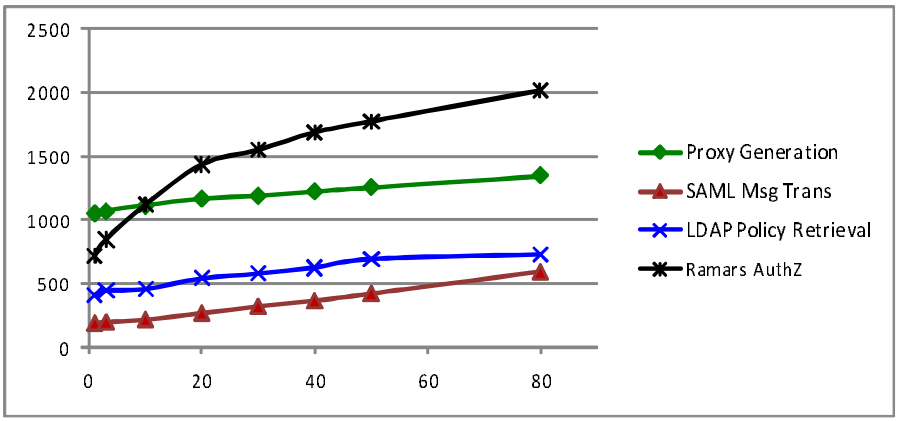
4 System evaluation

We conduct a series of experiments to evaluate how well the system scales along with the increased evaluation complexity and also analyse the overhead of the application of RamarsAuthZ authorisation service over Globus DRS service. In our testbed, the enhanced DRS service and RAMARS adaptor are deployed within Globus toolkit version 4.0.5 on a Pentium IV machine with Fedora Core 6. The *Ramars-proxy-init* tool is also deployed at the same machine for a Grid client to generate RamarsAuthZ proxy certificates and invoke the DRS service. The RamarsAuthZ service is employed within a WSRF-compliant WS Java container on a Pentium M machine with Windows XP. The Shibboleth IdP 1.3.3 is installed on a Pentium IV machine with Fedora Core 4. And IPlanet directory server is installed on a Pentium IV Windows XP machine as the back-end LDAP repository for the originator's ROA policies. All machines are located within the University's domain. We develop metrics to evaluate the performance of the system and the measurement of these metrics is performed by applying monitors at various locations of the system. Figure 4(a) illustrates our testbed and the monitors we put in our system based on the following metrics.

Figure 4 RAMARS system evaluation (a) RamarsAuthZ system evaluation testbed (b) RamarsAuthZ system performance evaluation result (see online version for colours)



(a)



(b)

- *Attribute retrieval time* (T_{attr}): T_{attr} is the time taken by a Grid client to retrieve his attribute assertions from Shibboleth IdP.
- *Proxy generation time* (T_{proxy}): T_{proxy} is the time taken by a Grid client to invoke $Ramars - proxy - init$ tool to embed attribute assertions and Shibboleth IdP information within a proxy certificate.
- *DRS invocation time* (T_{invo}): T_{invo} is the time taken by a Grid client to invoke the enhanced Globus DRS service for completing a file replication task.
- *SAML message transfer time* (T_{saml}): T_{saml} is the time taken by the Globus DRS service to send out an authorisation request to RamarsAuthZ service and get the response back over SAML message exchange protocol.

- Policy retrieval time (T_{policy}): T_{policy} is the time taken by the RamarsAuthZ service to retrieve an originator's ROA policies from LDAP policy store.
- Policy evaluation time (T_{eval}): T_{eval} is the time taken by the RamarsAuthZ service to make an authorisation decision based on the originator's ROA policies.

As a Grid client's authorisation privileges are determined by his attributes, the client's attribute assertions need to be transferred all the way from the client through the Globus DRS service to the RamarsAuthZ service for making authorisation decisions. In this sense, the scalability of the system is largely affected by the number of attribute assertions handled by the system. Therefore, our first experiment is conducted by increasing the number of attribute assertions. Figure 4(b) indicates the process time in miliseconds as the number of attribute assertions gradually increases. In particular, SAML message transfer time T_{saml} and proxy generation T_{proxy} increase linearly as they have direct associations with the size of attribute assertions. The increase of attributes, however, has an indirect effect on the size of the originator's ROA policies. Therefore, LDAP policy retrieval T_{policy} increases with a flatter rate. The RamarsAuthZ policy evaluation T_{eval}, on the other hand, shows a polynomial trend with the increase of policy evaluation complexities, which is a desirable property that makes the system more scalable to a large number of attributes.

Our next experiment is conducted to analyse the overhead of RamarsAuthZ with the Globus DRS data sharing service. In particular, not only the DRS service itself needs to be authorised, but also the data replication operations require a series of fine-grained authorisations for each step of data replication. The overhead introduced to achieve such fine-grained authorisations should be measured. Meanwhile, the application of standard SAML authorisation message protocols need further evaluations through explorations and experiments. According to a typical scientific collaboration of DØ experiment (Fermi National Accelerator Laboratory, 2007), 300 DØ users submitted 15,000 requests involving 2-4TB data transfer per day with an average of 130 M bytes data transfer per query. We choose a sample data file of size 121,781 KB to be replicated in Globus DRS service for our experiment. The base time for DRS invocation time T_{invo} is measured by applying the default GridMap authorisation. We deploy RamarsAuthZ as a remote SAML-enabled authorisation service to measure the overhead of the one-step authorisation for DRS service, and then measure the fine-grained multiple-step authorisations for DRS operations. In addition, to better understand the overhead of standard SAML authorisation protocols, we deploy RamarsAuthZ as a local authorisation module so that SAML message exchange is replaced by a local procedure call. Table 1 shows the results of our experiment when we adjust the number of attributes from 1 to 80. The results are measured in milliseconds and computed based on the average of 100 test runs. With the extreme complexity of evaluation, the one-step RamarsAuthZ authorisation for DRS service introduces less than 7% overhead compared to the traditional GridMap authorisation, which we believe is a promising outcome with respect to the performance of RamarsAuthZ authorisation service. Same to our expectation, achieving fine-grained authorisation for stepwise data sharing involves considerable cost. However, considering the potential reduction of the administrative overhead against the practices of manually maintaining individual user accounts, RamarsAuthZ service still shows clear advantages both architecturally and technologically. Compared to the locally deployed authorisation module, the overhead of SAML authorisation messages cannot be neglected. Therefore, the usage of SAML

for authorisation in Grid systems needs to be limited for simple and optimised message assertion exchanges. Especially, instead of transferring a large number of attribute assertions as 'push' mode, a reference to the Grid client's IdPs should be transferred within SAML message for RamarsAuthZ to operate under 'pull' mode.

Table 1 RamarsAuthZ overhead analysis

Attr #	*Base*	*Remote for DRS service*		*Remote for DRS operations*		*Local module*	
		Time	*Overhead*	*Time*	*Overhead*	*Time*	*Overhead*
1	42584	1326	2.91%	45614	7.12%	44834	5.28%
3	42584	1497	3.26%	45948	7.90%	45132	5.98%
10	42584	1804	3.87%	46598	9.43%	45706	7.33%
40	42584	2688	5.54%	48527	13.96%	47043	10.47%
80	42584	3347	6.66%	50256	18.02%	47856	12.38%

In order to further justify our approach, we compare our RamaraAuthZ system with a number of existing authorisation systems for Grid environments, including VOMS (Groeper et al., 2007), Akenti (Thompson et al., 1999) and Permis (Chadwick et al., 2006). According to the access control challenges as we identified in the Introduction, we compare the systems from the following aspects:

- Originator control: to examine whether the authorisation system provides facility for the data resource owner to control the access.
- Attribute-based access control: to examine whether the authorisation system establishes authorisation based on user attributes.
- Policy engine: to examine whether the authorisation system consists of a policy engine for complex policy evaluations.
- Policy store: to examine the operation mode of the policy store for the authorisation system.
- Supported IdPs: to examine the number of IdPs supported by the authorisation system.
- Policy and IdP configuration: to examine whether the authorisation system relies on static configurations for policy and attribute management.
- Operation modes: to examine the policy and attribute retrieval mode of the authorisation system.
- Service-level control: to examine whether the authorisation system supports service-level control to protect Grid services.
- Data-level control: to examine whether the authorisation system supports data-level control to protect data resources being process by Grid data sharing services.

Table 2 summarises comparisons between our proposed RamarsAuthZ system with the above-mentioned authorisation systems. It is evidently shown that RamarsAuthZ system has advantages in various aspects such as established access control model and policy, system architecture and deployment, and the protection ranges.

Table 2 Comparison with related work

	VOMS with Shibboleth	*Akenti with Shibboleth*	*Permis RamarsAuthZ*	*RamarsAuthZ*
Originator control	Partial	Yes	Partial	Yes
Attribute-based access control	Yes	Yes	Yes	Yes
Policy engine	No	Yes	Yes	Yes
Policy store	Centralised, single	Distributed, multiple	Centralised, single	Distributed, multiple
Supported IdPs	Multiple	Multiple	Single	Multiple
Policy and IdP configuration	Static	Static	Static	Dynamic
Operation modes	Push	Pull	Pull	Push, pull
Service-level control	Yes	N/A	Yes	Yes
Data-level control	Maybe	N/A	No	Yes

5 Conclusions

In this paper, we proposed an integrated solution that provides effective policy-driven role-based access control for both Grid services and data resources. The RamarsAuthZ service does not rely on centralised policy stores and attribute authorities, which increases the scalability and portability of the service to serve the authorisation functionalities for various Grid services. The support for hybrid 'push' and 'pull' modes achieves great flexibility to meet the users' requirements. We also shared our experience in designing and building a proof-of-concept system. As our future work, our preliminary testing has indicated that SAML communications and policy retrievals are two important factors affecting the overall system performance. Therefore, caching mechanisms can be implemented within the RamarsAuthZ service to reduce the number of such communications for policy evaluation. In addition, we would further explore mechanisms for RamarsAuthZ to negotiate with Shibboleth IdP for retrieving only necessary attributes so that the authorisation decision can be derived more efficiently. Also, we would articulate a sophisticated and lightweight authentication protocol that can be interoperable with RamarsAuthZ. Identity-based approach (Hongwei et al., 2008) would help design an efficient and convenient security service without demanding any additional infrastructure- or domain-specifc protocol.

Acknowledgements

This work was partially supported by the grants from National Science Foundation and Department of Energy.

References

Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, A., Gianoli, A., Lorentey, L. and Spataro, F. (2003) 'VOMS, an authorization system for virtual organizations', *Proc. of 1st European Across Grids Conferences*.

Antonioletti, M., Atkinson, M., Baxter, R., Borley, A. and Hong, N.P.C. et al. (2005) 'The design and implementation of Grid database services in OGSA-DAI', *Concurrency and Computation: Practice and Experience*, Vol. 17, pp.357–376.

Cantor, S. (2005) *Shibboleth Architecture, Protocols and Profiles*, available at http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-latest.pdf.

Chadwick, D.W., Novikov, A. and Otenko, A. (2006) 'GridShib and Permis integration', *Campus-Wide Information Systems*, Vol. 23, No. 4, pp.297–308.

Fermi National Accelerator Laboratory (2007) *The DZero Experiment*, available at http://www-d0.fnal.gov/.

Foster, I. (2006) 'Globus toolkit version 4: software for service-oriented systems', *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag, LNCS 3779, pp.2–13.

Foster, I., Kesselman, C., Nick, J. and Tuecke, S. (2002) 'The physiology of the Grid: an open Grid services architecture for distributed systems integration', *Open Grid Service Infrastructure WG, Global Grid Forum*.

Globus Alliance (2006) *Globus Toolkit*, available at http://www.globus.org/toolkit/.

GridShib Project (2008) *Gridshib: A Policy Controlled Attribute Framework*, available at http://Gridshib.globus.org/.

Groeper, R., Grimm, C., Piger, S. and Wiebelitz, J. (2007) 'An architecture for authorization in Grids using Shibboleth and VOMS', *Proc. of 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*.

Hongwei, L., Shixin, S. and Haomiao, Y. (2008) 'Identity-based authentication protocol for Grid', *Journal of Systems Engineering and Electronics*, Vol. 19, No. 4, pp.860–865.

Housley, R., Polk, W., Ford, W. and Solo, D. (2002) *Internet x.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC4280, available at http://rfc.net/rfc3280.html.

Jin, J. and Ahn, G-J. (2006) 'Role-based access management for ad-hoc collaborative sharing', *Proc. of 11th Symposium on Access Control Models and Technologies (SACMAT)*.

Jin, J., Ahn, G-J., Shehab, M. and Hu, H. (2007) 'Towards trust-aware access management for ad-hoc collaborations', *Proc. of 3rd IEEE International Conference on Collaborative Computing*.

Laccetti, G. and Schmid, G. (2007) 'A framework model for Grid security', *Future Gener. Comput. Syst.*, Vol. 23, No. 5, pp.702–713.

Lang, B., Foster, I., Siebenlist, F., Ananthakrishnan, R. and Freeman, T. (2008) 'A flexible attribute based access control method for Grid computing', *Journal of Grid Computing*, Vol. 7, No. 2, pp.169–180.

OASIS (2003) *Assertions and Protocol for the Oasis Security Assertion Markup Language (SAML) v1.1.*, available at http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf.

OASIS (2005) *XACML 2.0 Core: Extensible Access Control Markup Language (XACML) Version 2.0.*, available at http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

Rajasekar, A., Wan, M. and Moore, R. (2002) 'MySRB and SRB – components of a data Grid', *Proc. of International Symposium on High Performance Distributed Computing (HPDC)*.

Raman, V., Narang, I., Crone, C., Haas, L., Malaika, S., Mukai, T., Wolfson, D. and Baru, C. (2002) *Data Access and Management Services on Grid*, available at http://www.nesc.ac.uk/talks/ggf5_hpdc11/damsg220702.pdf.

Thompson, M., Johnston, W., Mudumbai, S., Hoo, G., Jackson, K. and Essiari, A. (1999) 'Certificate-based access control for widely distributed resources', *Proc. of 8th Usenix Security Symposium*.

Tuecke, S., Welch, V., Engert, D., Pearlman, L. and Thompson, M. (2004) *Internet x.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*, available at http://rfc.net/rfc3820.html.

Welch, V., Ananthakrishnan, R., Siebenlist, F., Chadwick, D., Meder, S. and Pearlman, L. (2006) *Use of SAML for OGSI Authorization*, available at http://www.ggf.org/documents/GFD.66.pdf.

Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L. and Tuecke, S. (2003) 'Security for Grid services', *Proc. of 12th IEEE International Symposium on High Performance Distributed Computing*, pp.48–57.