

# Experiences in Teaching Grid Computing to Advanced Level Students

Dr R.O. Sinnott, A.J. Stell, Dr J. Watt  
National e-Science Centre  
University of Glasgow  
[ros@dcs.gla.ac.uk](mailto:ros@dcs.gla.ac.uk)

## Abstract

The development of teaching materials for future software engineers is critical to the long term success of the Grid. At present however there is considerable turmoil in the Grid community both within the standards and the technology base underpinning these standards. In this context, it is especially challenging to develop teaching materials that have some sort of lifetime beyond the next wave of Grid middleware and standards. In addition, the current way in which Grid security is supported and delivered has two key problems. Firstly in the case of the UK e-Science community, scalability issues arise from a central certificate authority. Secondly, the current security mechanisms used by the Grid community are not fine grained enough. In this paper we outline how these issues are being addressed through the development of a Grid Computing module supported by an advanced authorisation infrastructure at the University of Glasgow.

## 1. Introduction

Future Grid engineers require training materials that allow them to understand the way in which current Grid know-how has been delivered, its limitations, and importantly to understand how such middleware can be used to support e-Science today. We distinguish understanding the principles and challenges associated with the development of Grid technology and its general usage. The former we regard here as education, whilst the latter can be seen more generally as training. This distinction is important to make since the underlying challenges faced by internet wide heterogeneous distributed systems as addressed by Grid technology remains predominantly the same. The way in which existing software has evolved to meet these challenges is changing however.

In a similar vein, the distinction between Grid usage and e-Science more generally should be made since the borderline between using Grid middleware and undertaking e-Science is often blurred. It is clear that what we would like to achieve is: *to educate future computer scientists to engineer improved Grid middleware and to educate e-Scientists to use existing middleware to solve scientific problems*. We note here that e-Science education and training does not explicitly require or advocate use of Grid technology. Indeed there are many e-Scientists currently undertaking their research with

minimal or no Grid infrastructure. In short, the requirements for education and training differ depending upon who the target audience is.

At the time of writing, one of the greatest challenges in delivering materials for educating and/or training future Grid engineers and e-Scientists is the fluidity of the technological landscape. Grid technology and associated standards are perpetually evolving with new recommendations and software from standards bodies and solutions providers. This has been exemplified in the last year with the move from Grid infrastructures to Open Grid Service Infrastructure (OGSI) based Grid services and the move towards Web Service Resource Framework (WSRF) web/Grid services. The evolution of the Open Grid Service Architecture (OGSA) is also a key issue that makes the development and delivery of any form of education or materials difficult. Trainers and educators need to be sure that they are developing materials which has some expectancy of life time. Developing and delivering educational materials based upon explicit technology, e.g. Globus toolkit version 3, are fraught with dangers associated with a moving technology base.

## 2. Virtual Organisations and Security

One of the primary motivations for using or developing Grid middleware today is to

*dynamically* link collections of distributed individuals and resources together to form so called *Virtual Organisations* (VOs). Typically a VO will allow a collection of individuals and or institutions to pool resources such as data sets, data archives, CPUs, or allow access to specialised equipment from astronomical radio-telescopes through to medical imaging scanners. With the open and collaborative nature of the Grid, ensuring that local security constraints are met and not weakened by Grid security solutions is paramount. Public Key Infrastructures (PKIs) represent the most common way in which security in the Grid is addressed. Through PKIs, it is possible to validate the identity of a given user requesting access to a given resource. This is known as *authentication*. For example, with the Globus toolkit [12] solution, *gatekeepers* are used to ensure that signed requests are valid, i.e. from known collaborators. When this is so, i.e. the Distinguished Name (DN) of the requestor is in a locally stored and managed *gridmap* file, the user is typically given access to the locally set up account as defined in the *gridmap* file.

There are several key limitations with this approach with regard to security however. For example, the level of granularity of security is limited. There is no mention of what the user is allowed to do once they have gained access to the resource. Further this approach works on the assumption that user certificates are provided by an acknowledged certificate authority (CA). In the UK, a centrally managed CA at Rutherford Appleton Laboratories exists which (necessarily!) has strict procedures for how certificates are allocated. Users are expected to “prove” who they are in order to get a certificate, e.g. through presenting their passports to a trusted individual at their institution. This is a human intensive activity and one which has scalability issues once it is rolled out to the wider community, e.g. to industry and larger groups such as students taking Grid/e-Science courses. Having users personally take care of their private keys is another limitation of this approach.

In short, current experiences with PKIs [13, 14] as the mechanism for ensuring security on the Grid have not been too successful [15, 16]. Authorisation infrastructures offer extended and finer grained security control when accessing and using Grid resources. Numerous technological solutions have been put forward providing various levels of authorisation capabilities e.g. AKENTI [1], CAS [2],

CARDEA [3], GSI [4], PERMIS [5,6,7] and VOMS [8,9]. Examples of how these compare to one another are described in [17, 18, 19]. It is too early to say if large scale use of attribute certificates (ACs) for user authorisation, based on infrastructures such as PERMIS, will be successful or not. However, few other alternatives currently exist, so practical experience is required. In order for large scale use to be facilitated, dynamic (rather than static) delegation of authority is required. In the current PERMIS infrastructure, static delegation of authority means that a central authority has to be contacted, and register local managers in its policy, before managers are entitled to assign privileges to subordinates. With dynamic delegation of authority, local managers do not need to be registered, but are given the privilege to delegate when they are first given privileges to use the system. Managers can then allocate privileges to staff and students as required, without having to contact the central authority first to get permission. Through this, a federated and scalable model of security authorisation can be realised. In developing this federated Privilege Management Infrastructure (PMI) model, key challenges have to be overcome which are common to most, if not all, uses of Grid technology – the dynamic establishment of VOs. VOs allow shared use of computational and data resources by collaborating institutions. Establishing a VO will require that efficient access control mechanisms to the shared resources by known individuals are in place. However, currently in the Grid community access control is usually done by comparing the authenticated name of an entity to a name in an Access Control List. This approach lacks scalability and manageability as discussed in [15]. Dynamic delegation of privileges offers a more realistic approach that could shape future Grid security, especially when it is rolled-out to the masses, e.g. Grid students, industry.

### **3. Teaching Principles Underlying the Grid**

In educating the future generation of Grid engineers, a balance between concepts and principles associated with Grids and e-Science is needed. Within the Grid Computing module at the University of Glasgow, the lecture material developed has been focused upon the underlying principles and challenges associated with Grid technologies. Thus whilst there might be

numerous technologies say for job scheduling (Condor, Sun Grid Engine, OpenPBS, Maui, ...), the basic principles of job scheduling and the specific challenges of large scale, wide area job scheduling remain the same. It is worth noting that the Grid Computing module at Glasgow has 16 full time students registered. This has more than exceeded initial expectations for a brand new course and module.

Establishing a course based solely upon basic principles and challenges associated with Grid technologies applied to e-Science, is unlikely to be suitable for a full time advanced course however. Experiments and investigations using current state of the art in Grid technology are needed. At Glasgow this has been through looking towards Globus toolkit version 3 (amongst other technologies), however, the main point is that this technology has not provided the cornerstone of the educational material. Rather it has provided a vehicle through which many of the basic principles have been demonstrated. It is this perspective that underpins the difference between training and education more generally.

A key requirement on Grid education is a broad scope and balance. Grid technology touches on many areas from security, usability, job scheduling and data management etc and developing single courses attempting to provide a complete picture of Grid today needs to be targeted to the right audience. For example, whilst high level overviews of Grid can be provided say to undergraduate students, it is more likely the case that complete and detailed overview materials are best delivered to computer science students that have the necessary grounding in related materials. At Glasgow for example, numerous pre-requisites existed for students to select the Grid Computing module. For example, students had to be competent in Java, knowledgeable in internet technologies, have experience of distributed algorithms and systems, and done some work on databases. (As it happened numerous students did not meet all requirements hence extra lecture material had to be provided, e.g. on XML based technologies and standards and web services). The Grid computing module contents at Glasgow is given in Table 1.

Here the focus was primarily upon teaching the principles, concepts and overarching challenges present in Grids today. For example, the lectures

on security provided an overview of the challenges of making Grids secure including concepts such as authentication, authorisation, accounting, auditing, confidentiality, privacy, data integrity, and trust. Exploration of current Grid security mechanisms, e.g. PKI based authentication and GSI based individual service/user based authorisation was presented, with focus on the many open challenges to be addressed to ensure Grid security is robust and meets the needs of the different e-Science communities.

Week 1	Lecture 1	Introduction to Grid Computing
	Lecture 2	Scalability and Heterogeneity
Week 2	Tutorial 1	Discussion of seminal Grid papers
	Lecture 3	Open standards and architectures
	Lecture 4	Implementations of the Grid architecture
Week 3	Lecture 5	Web services
	Lecture 6	Resource discovery/information services
	Tutorial 2	Exploring web services technologies with GT3
Week 4	Lecture 7	Grid security concepts
	Lecture 8	Virtual organizations
	Lecture 9	Security in practise
Week 5	Tutorial 3	Discussion of Grid security papers/Lab
	Lecture 10	Job scheduling and management
	Lecture 11	Job scheduling and management
Week 6	Tutorial 4	Discussion of job scheduling papers
	Lecture 12	Workflow management
	Tutorial 5	Q & A on programming exercise
Week 7	Lecture 13	Data access, integration and management
	Lecture 14	Data provenance and curation
	Tutorial 6	Discussion of data management/provenance
Week 8	Lecture 15	Bulk Data Transfer
	Lecture 16	Peer-to-peer communication
	Tutorial 7	Discussion of networking papers
Week 9	Lecture 17	Tools for Collaboration
	Tutorial 8	Discussion on future of Grid Computing
	Lecture 18	The future of Grid Computing
Week 10	Lecture 19	Sample applications
	Lecture 20	Review of major concepts
	Tutorial 9	Q & A

Table 1: Grid Computing module contents

A key requirement on rolling out any form of Grid education is to establish the infrastructure needed for this. *This is non-trivial and the effort involved should not be underestimated.* Grid technology and dependencies between packages is still a challenge for installation and general usage. This includes getting students equipped for working on the Grid, e.g. through X.509 certificates (issued from the UK e-Science certificate authority at RAL), or as was the case at Glasgow, establishing a local certificate authority.

For the Grid computing module at Glasgow, the following list of software for the Grid Computing Module was used. This software was initially built on a single machine and the disk image duplicated across the training cluster.

- PERMIS Privilege Allocator 1.5+, PERMIS API 1.3 requires:
  - Java SDK 1.4.2+ (includes JNDI)
  - IAIK JCE (contained in Privilege Allocator 1.5)
- Globus Toolkit 3.3 requires:
  - Apache Ant 1.6+
  - Junit 3.8.1
  - gcc 3.3+, YACC or Bison, GNUtar<sup>1</sup>
- Condor 6.6.5
- OGSA-DAI 4.0
- Tomcat 4.1.24+ (with GT3.3)
- JDBC Database (Postgresql)

In addition, a lab server was set up with OpenLDAP 2.1 for the attribute certificates generated by the PERMIS Privilege Allocator.

#### **4. Advanced Authorisation Infrastructures**

In a Grid environment, authentication (being able to establish the identity of a user) should be augmented with authorisation capabilities, which can be considered as what Grid users are allowed to do on a given Grid end-system. Thus “what users are allowed to do” can be interpreted as the privileges that the users have been allocated on those end-systems. The X.509 standard [20] has standardised the certificates of a PMI. A PMI can be considered as being related to authorisation in

much the same way as a PKI is related to authentication. Consequently, there are many similar concepts in PKIs and PMIs. An outline of these concepts and their relationship are discussed in detail in [6].

A key concept from PMI are attribute certificates (ACs) which, in much the same manner as public key certificates in PKI, maintain a strong binding between a user’s name and one or more privilege attributes. The entity that digitally signs a public key certificate is called a Certification Authority (CA) whilst the entity that signs an AC is called an Attribute Authority (AA). The root of trust of a PKI is sometimes called the root CA – which in terms of the UK e-Science community is given by the Grid Support centre at RAL [21]. The root of trust of the PMI is called the source of authority (SOA). CAs may have subordinate CAs whom they trust and to which they delegate the powers of authentication and certification. Similarly, SOAs may delegate their power of authorisation to subordinate AAs. If a user needs to have their signing key revoked, a CA will issue a certificate revocation list. Similarly, if a user needs to have authorisation permissions revoked, an AA will issue an attribute certificate revocation list (ACRL). Typically, a given users’ access rights are held as access control lists (ACLs) within each target resource. In an X.509 PMI, the access rights are held within the privilege attributes of ACs that are issued to users. A given privilege attribute within an AC will describe one or more of the user’s access rights. A target resource will then read a user’s AC to see if they are allowed to perform the action being requested.

The international standard | ITU-T recommendation X.812 [22] identifies two key components to support authorised access to a target (software/data resource to be protected): an Access control Enforcement Function (also known as a Policy Enforcement Point (PEP)) and an Access control Decision Function (also known as a Policy Decision Point (PDP)). The PEP ensures that all requests to access the target are authorised through checking with the PDP. The PDP’s authorisation decision policy is often represented through collections of rules (policies), e.g. stored in a Lightweight Directory Access Protocol (LDAP) server.

The different authorisation infrastructures associated with Grid technology have put forward their own mechanisms for realising

---

<sup>1</sup>Included in the Linux version used in the lab.

PEPs and PDPs. Recently however, the GGF has put forward a generic API – the SAML AuthZ API - which in principle provides a generic PEP that can be associated with an arbitrary authorisation infrastructure [23]. The Grid specification is an enhanced profile of the OASIS Security Assertion Markup Language v1.1 [24]. The OASIS SAML specification defines a number of elements for making assertions and queries regarding authentication, authorization decisions and attributes. The OASIS SAML AuthZ specification defines a message exchange between a PEP and a PDP, consisting of an *AuthorizationDecisionQuery* flowing from the PEP to the PDP, with an assertion returned containing some number of *AuthorizationDecisionStatements*.

The *AuthorizationDecisionQuery* consists of:

- A *Subject* element containing a *NameIdentifier* specifying the initiator identity
- A *Resource* element specifying the resource to which the request to be authorized is being made.
- One or more *Action* elements specifying the actions being requested on the resources

Through this SAML AuthZ API, a generic PEP can be achieved which can be associated with arbitrary (GT3.3) Grid services. Thus rather than developers having to explicitly engineer a PEP on a per application basis, the information contained within the deployment descriptor file (.wsdd) when the service is deployed within the container, is used. Authorisation checks on users attempting to invoke “*methods*” associated with this service are then made using the information in the .wsdd file and the contents of the LDAP repository (PDP) together with the DN of the user themselves. Note that this “method” authorisation basis extends current security mechanisms such as GSI which work on a per service/container basis. This generic solution can be applied to numerous infrastructures used to realise PDPs such as PERMIS.

#### 4.1 PERMIS Background

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) project [7] was an EC project that built an authorisation infrastructure to realise a scalable X.509 AC based PMI. Through PERMIS, an alternative and more scalable approach to

centrally allocated X.509 public key certificates can be achieved through the issuance of locally allocated X.509 ACs.

The PERMIS software realises a Role Based Access Control (RBAC) authorisation infrastructure. It offers a standards-based Java API that allows developers of resource gateways (gatekeepers) to enquire if a particular access to a resource should be allowed. The PERMIS RBAC system uses XML based policies defining rules, specifying which access control decisions are to be made for given VO resources. These rules include definitions of: subjects that can be assigned roles; SOAs, e.g. local managers trusted to assign roles to subjects; roles and their hierarchical relationships; what roles can be assigned to which subjects by which SOAs; target resources, and the actions that can be applied to them; which roles are allowed to perform which actions on which targets, and the conditions under which access can be granted to roles.

Roles are assigned to subjects by issuing them with X.509 Attribute Certificate(s). A graphical tool called the Privilege Allocator (PA) has been developed to support this process. Once roles are assigned, and policies developed, they are digitally signed by a manager and stored in one or more LDAP repositories.

The process to set up and use PERMIS can be split into two parts: *Administration* and *Use*. To set up and administer PERMIS requires the use of a LDAP server to store the attribute certificates and reference the SOA root certificate. A local CA is required to be set up using OpenSSL [25] – this designates the SOA and all user certificates created from this CA must have a Distinguished Name that matches the structure of the LDAP server. The DN of the user certificate is what is used to identify the client making the call on the grid service.

From the user’s perspective, once the administrator has set up the infrastructure, the PERMIS service is relatively easy to use. Unique identifiers are placed as parameters into the user’s grid service deployment descriptor (.wsdd file). These are the Object Identification (OID) number of the policy in the repository, the URI of the LDAP server where the policies are held and the SOA associated with the policy being implemented. Once these parameters are input and the service is deployed, the user creates a

proxy certificate with the user certificate created by the local CA to perform strong authentication. The client is run and the authorisation process allows or disallows the intended action.

## 5. Experiences of Grid Computing Module

Establishing lectures and associated tutorials encompassing a broad range of Grid and e-Science experiences and insight for the first time is a challenge. To help minimise this difficulty, the Grid computing module at Glasgow was developed through a combined effort including lecturing and tutorial duties (Drs Perkins, Sinnott and Watt) with invited lectures provided by Prof. Seamus Ross of the National Digital Curation Centre and Dr David Ferguson of the EGEE training team. Dr Watt was largely responsible for establishing the infrastructure used to explore Grid technologies and Anthony Stell for expertise in and deployment of security policies.

The course itself has just completed and students given positive comments on its content and delivery.

As part of the course, the students were asked to develop a policy through the PERMIS Policy Editor. This policy was then used in the larger programming assignment. Specifically the policy was for a GT3.3 service (*searchSortGridService*) which wraps a Condor based application (this service offers two methods to search (*searchMethod*) and sort (*sortMethod*) a large (5MB) text file (the complete works of Shakespeare). The students themselves were split into groups (*studentteam1*, *studentteam2*) with the authorisation policy to ensure that method *sortMethod* can **only** be invoked by members of their student group and the lecturing staff, and that method *searchMethod* can be invoked by everyone. This set-up was used to illustrate the use of Role-Based Access Control (RBAC), where users are allocated privileges based on what role they have been assigned rather than their local user credentials.

The output of the Policy Editor is an XML-based policy (Figure 1) which identifies specific roles (*studentteam1*, *studentteam2* and *lecturer*), specific targets (*searchSortGridService*) and specific actions on that target (*searchMethod* and *sortMethod*). This XML policy is then input to the Privilege Allocator tool which is used to denote specific users associated with that given

rule (i.e. the students themselves) and to digitally sign the policy and store it in the LDAP server.

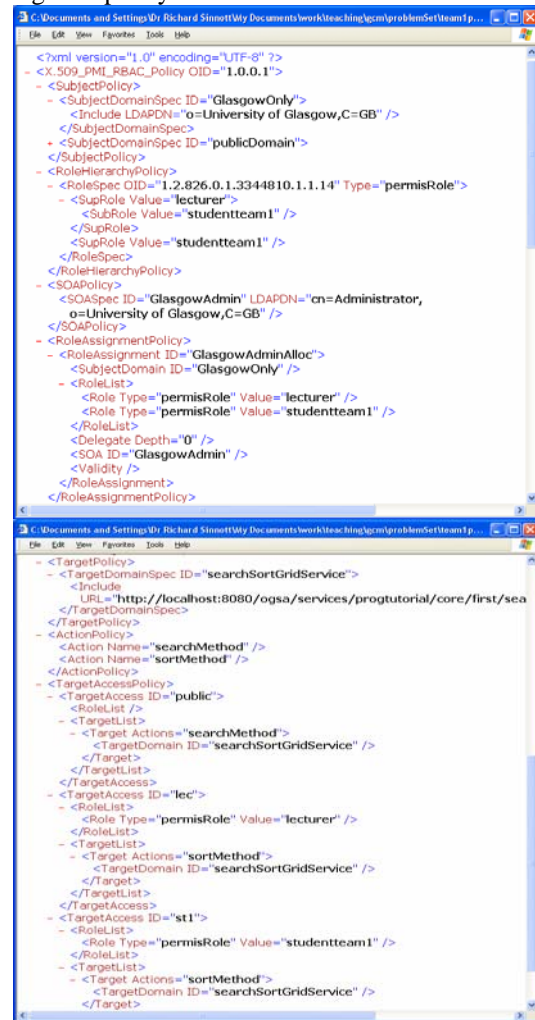


Figure 1: XML based security policy

All of the students were able to successfully create the policy shown above using the PERMIS Policy Editor with minimal help from staff. It should be noted that the students were informed of various background information that they would need to create the policy including the Policy Domain; the Source of Authority to use; and the Policy Object Identifier for their student group.

The policy was signed by the administrator of the LDAP server and input to the SOA node. Two policies were used; identical in every respect except for which team could access the restricted method. The students were split into two teams with each team having a specific policy identification number. The students were requested to critically evaluate the PERMIS tools for this purpose, with these results being sent

back to the PERMIS team for HCI improvements and minor bug fixes, e.g. problems in cross platform (Unix/Windows) versions of the tool and functionality in the tool that has not yet been implemented (although the buttons/pull down menus exist). This information has provided the most detailed exploration of PERMIS to date and is guiding future work on the PERMIS tools.

The programming assignment that was used to investigate Grid based solutions was based upon exploring various performance aspects of existing Grid middleware. Specifically the students were to write a Java based application to perform two operations:

- search a text file (and retrieve the number of times a certain term occurs);
- sort the text file so that the terms and number of times they occur is given.

Students were allowed to implement whatever search/sorting algorithm they wished. Students were then expected to:

- perform benchmarking on the speed of the application on a single PC for searching and sorting the text file;
- extend and parallelise the application to make use of the training lab Condor pool and perform benchmarking;
- wrap the parallelised application as a GT3.3 Grid service and develop a client to test it and perform benchmarking;
- extend the GT3.3 based parallelised application so that it uses the previously developed security policy;
- extend the GT3.3 based parallelised application so that it supports individual security restrictions (can user X invoke *searchSortGridService*) through GSI.

To allow the students to use the PERMIS infrastructure, they were issued with their own individual certificates and keys, generated from a local CA.

At the time of writing, the student implementations of this assignment have *just* been completed. It is clear that this has proven to be a challenging assignment with some students being able to get further than others. It should be noted that certain students did manage to complete their assignments. The overall performance benchmarking of the impact on distributing a search/sort job across a Condor

pool and delivering this as an authorised Grid service (using PERMIS) is significant however.

## 6. Conclusions and Future Plans

The Grid Computing module at Glasgow University has provided a broad overview of the key concepts and challenges associated with Grids. The exploration of these concepts through existing Grid middleware such as GT3 and Condor has proven to be a challenge, both for the educators and the students themselves. It goes without saying that the existence of a more mature Grid middleware and associated standards would have significantly reduced the overheads in preparing this module and for flattening the student learning curve.

The lecture material itself has been developed specifically for advanced Computer Scientists and hence makes various assumptions on their backgrounds. The lecture material itself is available for download (see <http://csperskins.org/teaching/2004-2005/gc5>) however it has a focus of advanced Grid education as opposed to general Grid training. As such, the material is focused on challenges of Grid technology as opposed to being more general training material.

This work has also allowed exploration of advanced authorisation infrastructures. The existing model of a single central CA for certificates used in PKI based authentication is not a scalable one, nor does it meet the challenges of Grid security. A more realistic model would be to have local CA infrastructure to issue certificates, e.g. to students as part of their matriculation. At Glasgow this has been relatively straightforward to achieve, however there are issues in recognition of these certificates by other CAs, e.g. the UK e-Science CA since no root of trust exists between these CAs. Solutions to this problem might be based upon some form of bridging technologies [26].

Whilst PERMIS and APIs such as the GGF SAML AuthZ API support a generic Grid based authorisation mechanism, there are likely to be scalability issues in such solutions. Thus for example, the expression of policies of individual methods on individual services to specific end users will have immediate maintenance and performance overheads when such systems are rolled out to hundreds or thousands of users accessing and using a multitude of services and

data sets. The data repository used for such infrastructures will also need to be enhanced as currently no tools exist that easily allow its dynamic maintenance.

Another issue that has been fed back to the PERMIS team and the wider authorisation standards community is the ability to have parameter-level authorisation (as opposed to the purely method name based authorisation supported by the SAML AuthZ API). This would allow for greater flexibility and finer grained authorisation possibilities.

In conclusion, there is a need for a core set of Grid and e-Science educational and training materials, as well as guidelines on how to establish local infrastructures for teaching purposes. Whilst as demonstrated here, it is possible to produce such materials and deploy the associated infrastructure, this is still challenging. With the eventual consolidation of standards and associated technologies, it is hoped that this situation improves.

## 6.1 Acknowledgements

The authors would like to thank collaborators from the PERMIS team (Prof David Chadwick and Dr Sassa Otenko) and the other Glasgow Grid Computing lecturer (Dr Colin Perkins). This work was supported by the JISC DyVOSE project [11] grant.

## 7. References

[1] Johnston, W., Mudumbai, S., Thompson, M. Authorization and Attribute Certificates for Widely Distributed Access Control, IEEE 7th Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA, June, 1998, p340-345 (<http://www-itg.lbl.gov/security/Akenti/>)

[2] L Pearlman, et al., A Community Authorisation Service for Group Collaboration, in Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. 2002.

[3] Lepro, R., Cardea: Dynamic Access Control in Distributed Systems, NASA Technical Report NAS-03-020, November 2003

[4] Globus Grid Security Infrastructure, <http://www-unix.globus.org/toolkit/docs/3.2/gsi/index.html>

[5] D.W.Chadwick, A. Otenko, E.Ball, Role-based Access Control with X.509 Attribute Certificates, IEEE Internet Computing, Mar-April 2003, pp. 62-69.

[6] D.W.Chadwick, A. Otenko, The PERMIS X.509 Role Based Privilege Management Infrastructure, Future Generation Computer Systems, 936 (2002) 1–13, December 2002. Elsevier Science BV.

[7] Privilege and Role Management Infrastructure Standards Validation project [www.permis.org](http://www.permis.org)

[8] VOMS Architecture, European Datagrid Authorization Working group, 5 September 2002.

[9] Steven Newhouse, Virtual Organisation Management, The London E-Science centre, <http://www.lesc.ic.ac.uk/projects/oscar-g.html>

[11] Dynamic Virtual Organisations in e-Science Education project (DyVOSE), [www.nesc.ac.uk/hub/projects/dyvo](http://www.nesc.ac.uk/hub/projects/dyvo)

[12] Globus, <http://www.globus.org>

[13] R. Housley, T. Polk, Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructures, Wiley Computer Publishing, 2001.

[14] ITU-T Recommendation X.509 (2001) | ISO/IEC 9594-8: 2001, Information technology – Open Systems Interconnection – Public-Key and Attribute Certificate Frameworks.

[15] JISC Authentication, Authorisation and Accounting (AAA) Programme Technologies for Information Environment Security (TIES), [http://www.edina.ac.uk/projects/ties/ties\\_23-9.pdf](http://www.edina.ac.uk/projects/ties/ties_23-9.pdf).

[16] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. Paper presented at the 9<sup>th</sup> USENIX security symposium, Washington, 1999.

[17] D. Chadwick, O. Otenko, A Comparison of the Akenti and PERMIS Authorization Infrastructures, in Ensuring Security in IT Infrastructures, Proceedings of ITI First International Conference on Information and Communications Technology (ICICT 2003) Cairo University, Ed. Mahmoud T El-Hadidi, p5-26, 2003

[18] Conceptual AuthZ Framework and Classification, [https://forge.gridforum.org/docman2/ViewCategory.php?group\\_id=55&category\\_id=458](https://forge.gridforum.org/docman2/ViewCategory.php?group_id=55&category_id=458)

[19] A.J. Stell, Grid Security: An Evaluation of Authorisation Infrastructures for Grid Computing, MSc Dissertation, University of Glasgow, 2004.

[20] ITU-T Rec. X.509 (2000) | ISO/IEC 9594-8. The Directory: Authentication Framework.

[21] UK e-Science Certification Authority, [www.grid-support.ac.uk](http://www.grid-support.ac.uk)

[22] ITU-T Rec X.812 (1995) | ISO/IEC 10181-3:1996, Security Frameworks for open systems: Access control framework

[23] V. Welch, F. Siebenlist, D. Chadwick, S. Meder, L. Pearlman, Use of SAML for OGSA Authorization, June 2004, <https://forge.gridforum.org/projects/ogsa-authz>

[24] OASIS. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1., 2 September 2003, <http://www.oasis-open.org/committees/security/>

[25] OpenSSL to create certificates, <http://www.flatmtn.com/computer/Linux-SSLCertificates.html>

[26] J. Jokl, J. Basney and M. Humphrey, Experiences using Bridge CAs for Grids, Proceedings of UK Workshop on Grid Security Practice - Oxford, July 2004