Programming Research Group

A CRITICAL SURVEY OF GRID SECURITY REQUIREMENTS AND TECHNOLOGIES

Philippa J. Broadfoot and Andrew P. Martin

PRG-RR-03-15

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

Abstract

Security in grid applications is a wide topic, touching many of the core issues in Information Security. We survey the distinctive features of grids from a security perspective, recording the principal security requirements which arise in some current projects we are engaged with, and in the literature. These include particular requirements on authentication, authorisation, revocation, confidentiality, distributed trust, integrity, and freshness. These requirements inform an analysis of existing technologies deployed in the European DataGrid, and we present an appraisal of the extent to which these meet the requirements presented. The paper includes an analysis of the key software engineering security issues which remain to be addressed.

1 Introduction

As pervasive and on-demand computing resources become a reality, the concept of a grid [FK99] is gaining popularity as a scalable way to deliver access to a wide range of distributed computing and data resources. These resources may span a broad range of heterogeneous and geographically dispersed applications, for example, distributed databases, distributed supercomputing, teleimmersion and specialist scientific instruments. Providing robust middleware to support such work — as distinct from particular distributed computing applications — represents a significant software engineering challenge.

Those challenges are particularly substantial in the area of security, which is the subject of our study. The rules which govern the sharing of resources must be clearly defined: both in terms of who can access what resource, and under what conditions. Grids have become particularly associated with the notion of a *Virtual Organisation* (VO) [FKT01]: a dynamic grouping of users and resources brought together for a particular task, often a short-lived one and operating under an agreed set of regulations. A major aim of the grid technology being developed over the last few years is to enable this type of architecture and provide the seamless sharing of resources within it.

Our purpose in writing is to identify the security issues that are distinctive within a grid environment (as distinct from the classic client-server model commonly used over the Internet). Some of these issues have been clearly highlighted in the grid literature (for example, [FKTT98]), but many of the subtleties remain unclear. It is our observation, in particular within the EU DataGrid project [Dat], that security poses substantial problems in the development and deployment of grid technology.

Secondly, we survey currently technology which has been proposed to address security within a grid environment. Much of our experience lies within the EU DataGrid project and we will therefore primarily focus on its technologies. We discuss to what extent security has been addressed within a grid environment (in particular within this project), and which areas remain unaddressed.

Research in grid technologies necessitates consideration of security as a whole-system property. This provides a useful counter-balance to the style of academic study which has concentrated on more narrow concerns such as protocol analysis: many formal security analysis methods have been developed and successfully used to find security flaws or prove their absence (for example [RSG$^+$00]). Correctness in that sense, and also as it relates to concepts such as confidentiality and authentication, is critical and now quite well-understood. Implementing grids serves to underline the remaining gap between such theory and practice.

In the following section, we record the characteristics of this kind of general-purpose grid solution, and follow that with a Section considering the distinctive grid security aspects arise. Section 4 describes the technologies which are available and proposed to meet these requirements. A final section considers the extent to which this has been achieved.

2 Distinctive characteristics of a grid

The purpose of a grid is to enable large scale, distributed resource sharing across dynamic, heterogeneous and geographically dispersed virtual organisations. The term 'grid' seems to be used for a wide range of artifacts. Some may appear to offer little more than a moderately scalable way to connect UNIX$^{\text{TM}}$ systems together; increasingly, however, a large collection of middleware services is envisaged.

Much of the development of grid solutions has been led by applications, and many security concerns have been suitably addressed in those contexts. Notably, the majority of these have been supercomputing applications with almost no confidentiality concerns, distributed in such a way as to raise few questions of integrity or availability. For other authors, the greatest interest is with the potential for highly distributed computing in the public resource model [KWA+01].

By contrast, a 'production grid' [Joh03] will offer a persistent service, capable of supporting widely heterogeneous tasks, with varying security needs. Moreover, the devices connected to such a grid will most likely include some of the fastest computing clusters and largest data stores in the world, connected by the highest speed network backbones. The scope for denial of service is substantial (both within such a grid, and its use to launch such an attack on the wider Internet) the possibility of surreptitious use, perhaps for nefarious purposes, is possibly more disturbing.

The notion of grid technology embraced by the Global Grid Forum [GGF], the UK eScience Programme, and others, is the one we will adopt — essentially one of any large-scale technology utilising the global Internet to build distributed computing and communications infrastructures.

Grids collect and generalise many long-standing distributed computing ideas. They are more complex than their precursors, but encompass many of the problems found in traditional networks as well as a class of new ones. In this section, we aim to identity the key characteristics that distinguish grid architectures from traditional networks; we will concentrate primarily on those features that have a direct impact upon security. From the perspective of security, grid solutions appear to be characterised by:

Dynamic network One of the aims of a grid is to enable the sharing of vast amounts of distributed resources within large, dynamic and distributed communities of users, where the availability of resources, membership of communities (or virtual organisations) and access rights are continually changing and evolving. A grid is expected to provide an architecture that enables such a dynamic structure.

These changing patterns of use add considerably to the already great challenge of allowing access to remote resources owned and managed by third parties: issues of trust and liability become very important. Most existing network services are free and almost anonymous at the point of use (for example, DNS, time servers and informational web sites), or else they enter into a simple relationship with the user (for example, e-Commerce and chat rooms).

Single sign-on A consequence of scale is that users may have access to a large range of resources, with differing patterns of availability, differing access controls, differing costs and so on. Users may access these via a number of different clients, depending on their modes of work or their need to orchestrate computations remotely. This problem affects users of systems other than grids, of course, but where the collection of resources to be used cannot be determined ahead of time, the need for a single means of identification becomes all the more pressing.

Hence, an important requirement within a grid environment is referred to as *single sign-on*, defined in [FKTT98] as: "A user should be able to authenticate once (for example, when starting a computation) and initiate computations that acquire resources, use resources, release resources, and communicate internally, without further authentication of the user." The idea is that a user can have access to a large number of distributed resources without having to actively authenticate himself to each one individually. A single job may involve many users and resources with computations running in parallel (as opposed to the traditional one-to-one client to server interactions). Furthermore, users need to be able to set computations running for long durations of time (for example, several weeks) without necessarily remaining logged on to their local system throughout.

Interoperability A community of users will support their own local policies regarding membership, access rights to resources, security and so on. Similarly, the sites providing resources will handle local issues such as availability, access control and security. A grid architecture is expected to enable the interoperability between sites and communities with differing local policies and ensure that the respective policies are adhered to in each case. Moreover, one may expect different Virtual Organisations to have differing security requirements: medical image processing and earth observation image processing may use very similar resources, but the confidentiality requirements will differ greatly.

The general instance of a grid will not only contain a large number of components, but will be characterised by considerable heterogeneity among these. Some may be simple storage elements, incorporating POSIX I/O operations; some may be computing elements, with various architectures, for example, supercomputers, clusters, differing processor designs; others may be databases and scientific instruments. Access to these will also be via heterogeneous clients, running under different kinds of operating systems.

The Web has achieved much in the area of interoperability in this sense, and has promoted many standards which help to realise for the Internet something approaching the ‘Presentation’ layer in the OSI reference model [DZ83]. Part of the grid vision is to achieve this for a much broader class of data and access to computation.

Scalability The scale of the solution tends to dominate a discussion of grids. A grid composed of just a handful of nodes and a small number of users could be designed, implemented and managed in a very simple way. A grid that is capable of reaching an extent equivalent to that of the World-Wide Web is a much larger engineering problem. It must be capable of easy expansion, graceful degradation, localised management and accounting, progressive replacement, fault recovery, dynamic optimisation, and so on.

Delegation To enable the sharing of vast amounts of data and computational resources across highly distributed communities of users effectively and efficiently (and to satisfy the single sign-on requirement), it is essential to enable delegation. One of the distinguishing factors required within a grid infrastructure is the necessity to delegate *entity rights* to other entities within the grid.

Consider the following example: A user U wishes to access some computational resource R, in order to run computationally intensive simulations over large sets of data that are stored on remote distributed databases. Furthermore, U grants R the ability to access the necessary data to enable the simulations to be carried. This is achieved by delegating a subset of U's rights, thereby allowing R to act on U's behalf for this particular job. This is one example of the sort of functionality required within a grid architecture. The ability and freedom to delegate user rights will depend upon the security requirements of the application.

Delegation is a well-known concept (for example, [GM90]), but has proved difficult to achieve securely in practice. As discussed in [BL03], enabling the delegation of user rights has a significant impact upon the overall security within a grid environment and raises many subtle issues to be resolved. This is especially acute when dealing with mechanisms that allow *delegation chaining*: A user U delegates rights to some entity E_1; then E_1 delegates these rights on to another entity E_2 and so on, forming a delegation chain.

3 Security aspects distinctive to a grid

Considering the kind of grid infrastructure and grid applications outlined above, we describe in this section the main distinctive security characteristics that emerge in this context. As before, we concentrate on those issues which are either unique in a grid context or have substantial novelty due to the grid features described above.

3.1 Authentication

The scale, complexity, and heterogeneity in a grid situation make the clean separation of authentication from authorisation very desirable. ‘Pure’ authentication schemes are surprisingly rare in everyday use, so grid solutions must be carefully designed.

The authentication requirements between two nodes within a grid infrastructure are not particularly distinct from those traditionally experienced in client-server networks. However, the main hurdles encountered when applying standard mechanisms such as Public Key Infrastructure (PKI) include scalability, trust across different Certification Authorities and timeliness of revocation (see Section 3.3 below). Although the technology for authentication is quite well-established, its integration into users' work patterns may not be. We highlight some of the issues that arise:

Key management The requirement that *end users* keep a private key secret is relatively unusual in many of the communities anticipating grid applications. Moreover, users may need to be able to access grid resources from a number of different hosts — seeming to require the use of some portable medium for key storage.

Furthermore, the challenges involved for users in key management are often compounded: for various reasons, users will often have several different credentials—to cover different roles, or because they are part of several projects which have elected to trust different Certification Authorities. Whilst public key infrastructure could handle this situation by signing the same public key into several different certificates, in practice the user may end up with numerous key pairs to manage.

There may be good reasons to link together these different identities; reasons of privacy and robustness in the presence of compromise urge caution. The issues are very similar to the concept of *federated identity* arising in Web applications, where two leaders are *Microsoft Passport* [Mic02] and *Liberty Alliance* [HW03].

Indirect authentication Many grid applications employ delegation to allow complex distributed patterns of action. Strictly, this is largely an issue of *authorisation*, but those processes with delegated authority appear to be acting for their 'owner' — hence a question of authentication. Where delegation is chained, authentication becomes more complex and no longer involves two nodes. What authentication needs to be established and between which components? If we allow a delegation chain to be created, where a user U's rights are delegated down a list of processes P_1 through to P_n respectively, who should be authenticated to whom? Should every process in the chain be authenticated to U and/or the final resource being accessed? These questions must be addressed when deploying delegation.

3.2 Authorisation

The potentially large numbers of users and resources with differing management and policies leads to an unusually sharp distinction between authentication and authorisation in a grid context. Authentication is concerned with the identity of an entity (for example, a user, service or resource) within a given context (for example, within a virtual organisation). On the other hand, given an entity with identity I, authorisation captures what I can access and do.

Whereas in operating system access mechanisms, the acquisition of a user account typically gives immediate general access to a wide range of resources, and whereas in many web-based solutions, a user account gives access to precisely one resource (or to a fixed menu of resources), in a grid resource owners will require the ability to grant or deny access based on identity, membership of groups or virtual organisations, credit-worthiness, and other dynamic considerations. A single grid entity may have multiple roles and be a member of multiple virtual organisations, resulting in varying levels of authorisation capabilities. The architecture must be such that a given user U can request access to resources that satisfy a given set of characteristics, as opposed to a specific resource known by U.

Scalability becomes an important factor when considering authorisation mechanisms within a grid environment, especially due to its dynamic nature. It would not, for example, be feasible to make resource providers maintain an access control list for every member within its virtual organisation and handle all the continuous changes of their accessibility rights throughout the lifetime of a project.

A final point that needs to be considered within a grid, is that the authorisation of data stored upon a particular resource is not necessarily bound to the actual resource itself, but rather to the data. This is particularly relevant when deploying replication of data for efficiency and scalability purposes. One needs to draw a distinction between the accessibility of a particular data element and the resources upon which it is stored. Enforcing consistency of access to replicated instances of data becomes vital.

3.3 Revocation

Revocation is vital for both authentication, for example, when a key is compromised, and authorisation, for example, when a user's project ends, a VO winds up or a user proves untrustworthy. The criticality of such mechanisms relies greatly on the application area: it is not uncommon for Universities to continue to allow access to systems for months, or even years after a member of staff has left. Conversely, in some business contexts, an instant withdrawal of access is effected. For a grid to be trustworthy it must support the latter — this is an important part of managability.

3.4 Confidentiality

The data being processed in a grid may be subject to considerable confidentiality constraints, either due to privacy concerns or issues of intellectual property: for example, existing and proposed grid applications involve patient data [BGS+03], bioinformatics and genomics databases [BB03] and industrial design information [GEO]. Confidentiality is frequently associated with the encryption of data only, for example, across a public medium such as the Internet. There are, however, other aspects to be considered:

Data on remote resources Whereas in non-grid applications, confidential data would be kept as far as possible from the Internet, the use of a grid implies that they are stored in online accessible databases. Access to their interfaces must be carefully controlled, both to allow access only to appropriate users, and also to allow queries and simulations to run over these highly confidential data without that data being compromised or revealed. In some cases, the very execution of a query may reveal secret information; if the database is to be shared in a grid, it might need to be operated by a trusted third party.

A further novelty of grid applications is that they may entail running confidential code or using confidential data on a remote resource: running a job on a dynamically-selected cluster according to load may be good resource management, but the data owner may know nothing about the trust status of the cluster selected by the grid software.

In some applications where the confidentiality of data is of utmost importance, it is not always desirable to rely solely upon access control mechanisms of the resource (for example, storage element) for protection against malicious intruders. Furthermore, for scalability and efficiency purposes within the grid, data is replicated and stored in multiple locations. Rather than relying on the trustworthiness of the resource, it would be wise to store the data encrypted. If the resource turns out to be malicious, does not provide adequate security measures or gets compromised, then the data will still be protected. For databases storing medical files, this extra level of security is essential. This leads to an interesting research avenue regarding the ability to run encrypted queries over encrypted databases.

User and resource privacy Confidentiality within a grid is not just concerned with data that is stored upon a resource; it also extends to the privacy requirements of the actual users and resources. Users are protected under privacy laws and these must be adhered to by all components of proposed grid technology. For example, it should not be possible for unauthorised entities to determine users' access rights, as this could lead to confidential data being deduced concerning a particular user. A grid component that is responsible and trusted to find available resources for a given user U with a set of resource requirements, must be able to collate this information. On the other hand, it does not, for example, need to know or be able to construct a list of resources where U has been banned. Clearly, the precise requirements will vary depending upon the application.

Deducibility One must ensure that any mechanism deployed to provide confidentiality of data also protects the deducibility of that data. This is an obvious requirement, but nonetheless one that is frequently ignored, leading to avoidable breaches of security.

Replication of data and consistency Due to the replication of data within a grid, one must ensure that the confidentiality of data remains consistent through replication. Furthermore, any dependencies between the mechanism of ensuring data confidentiality and its physical location must be identified and handled accordingly when replications are stored in different locations.

Law Countries have varying laws regarding privacy rights and encryption. When considering the deployment of grid technologies across international borders, for example, encryption mechanisms and the replication of confidential data, these need to be taken into account and adhered to.

3.5 Distributed trust

The trust status of a remote system or user is hard to determine — we may have no idea whether it has been compromised or indeed was malicious from the outset — so is the communications medium. Yet a grid must be constructed from such components, in a dynamic fashion. The hope is that a trustworthy system can be constructed from reasonable roots of trust: digital signatures attesting the identity of users and hosts.

Trust is of course a complex relationship: it is far from a binary relation (A trusts B); rather, a complex hierarchy is formed (A trusts B to do X and not to do Y). It is also closely bound up with — but broader than — authorisation.

Since the driving application of many grids — high-performance computing — requires users to be able to compile and run arbitrary code, this ability is built into that trust model, together with the almost inevitable ability to run an arbitrary shell command: users who are authorised to do something are able to do almost anything.

Such a model is inherently fragile, especially when combined with the desirable *single sign-on.* If a user's source of identification – for example the private key corresponding to a digital certificate – is disclosed, then an attacker gains very general and privileged access to a variety of systems and their network connections. Conversely, if a system is compromised (perhaps in manner not related to its role as a grid node), the damage is potentially even greater.

It is for precisely this reason that in the past system managers have generally aimed to keep trust domains relatively small, with few systems trusted by others; separate authentication to each system being preferred. Finer-grained authorisation models may help to limit this fragility.

When enabling delegation, determining trust relations between the participating components is essential. Some delegation mechanisms rely upon stringent trust requirements, without which most security aspects of the system will be compromised (for example, [GM90]). When a user U delegates a subset of its rights to some resource R, U must be satisfied that it trusts R. However, to what level does this trust extend to? Does U trust R to delegate U's rights further? Should R be allowed to delegate U's rights to anyone R trusts, anyone U trusts (or has stated) or no-one?

3.6 Integrity

Code and data on remote resources Many applications have strong integrity concerns: the data from some projects may not be confidential, but huge datasets from very expensive experiments cannot easily be re-generated. They must be beyond reproach. For data on the scale of those produced by the Large Hadron Collider (on the order of 1 Gbyte per second, continuously for many days [Seg00]), even verifying integrity (against a copy or a hash) would be a non-trivial task. This becomes a grid problem when copies or subsections of those data are managed automatically and stored at dispersed, separately managed locations.

Where data arises from remote processing or simulation, as with the issues of confidentiality, the trust status of the remote resource is of interest. Is the remote processor really running the intended code? To what extent should the accuracy of the results be trusted?

This problem is particularly acute in the area of public resource computing, such as climate*prediction*.net [SKM+02] where diverse members of the public are asked to run a simulation task. SETI@home has observed a number of users attempting to 'optimise' their code—often for altruistic reasons [ACK+02].

Delegation of access rights Integrity requirements also extend to the mechanisms by which users' rights are delegated: A malicious entity E (for example, a user or resource) cannot modify the set of rights that have been delegated to E by another user U.

3.7 Non-repudiation

Non-repudiation is concerned with the ability to prove that a given entity performed or agreed to a particular task, even in the face of denial. This is particularly important in e-commerce where money transactions take place. This is not a security aspect that has been considered in any detail within the grid, but it will become important as the grid technology matures and is used by applications involving financial exchanges. If accessing resources starts costing money, then both parties must be assured that the other fulfils their duty; in the cases where one party fails, proof of commitment is necessary to formally resolve the dispute.

3.8 Freshness

Freshness is closely related to authentication and authorisation and is frequently sidelined as a secondary concern. However, this is important in many grid applications:

Within a specified time frame One needs to be clear with regards to the validity of a user's proof of authentication and authorisation. In a traditional client-server protocol, this is straight forward. However, in a grid environment, user rights are delegated to enable more efficient access to resources and the duration of a job may vary from minutes to weeks. This notion of a time frame during which a particular user is authentic and authorised must be reflected in the proposed authentication and authorisation mechanisms.

Per usage control Some applications may have stringent access rules and require the ability to state precisely the number of times a given user may access a particular resource. This particularly becomes an issue when delegating users' rights. A user U may choose to delegate rights to another user U' and specify that U' may access resource R (on behalf of U) a specified number of times. There are numerous reasons for requiring this level of freshness, for example, where resources charge a per-access fee.

4 Technologies

In this section, we review the leading technologies which aim to meet the requirements detailed above. Our consideration is informed particularly by the European DataGrid, but we have sought to include representatives of all the significant technologies.

4.1 Authentication mechanisms

Public Key Infrastructure (PKI) PKI [Aus01] is deployed in a number of relatively narrow applications. Most Internet users encounter it (often unknowingly) in web site security where digital certificates are employed with SSL [Net96] to authenticate remote sites to web users. In this model of use, the authentication is one-sided, and trust in the certificates is bootstrapped from certificate issuers' self-signed certificates distributed with the browser or operating system.

By contrast, most grid implementations envisage PKI being deployed to the entire community of (grid) users. Technically, this presents no particular challenge, though it raises several questions relating to the number of Certificate Authorities (and subsidiary Registration Authorities) which may be required, and how they are to interrelate.

Specifically, PKI is employed in Globus [FKTT98], through its component called the grid Security Infrastructure (GSI). By implementing the Generic Security Service API (GSS-API) [Lin97], Globus is also able to interwork with Kerberos [NT94]. This provides interoperability with sites that deploy non-PKI authentication solutions.

Certificate Authority Policies The Policy documents produced by Certification Authorities — the Certification Practice Statement and the Certificate Policy — are intended to inform those relying on a certificate for identification of the level of trust which the CA expects to be possible. A grid node can determine, based on this information, whether to accept certificates signed by a given CA.

There has been much discussion of whether there should be many or few CAs. It would seem most natural that identity should be certified by or through the (real) organisation to which a user belongs (or, in the case of a member of the public, by a public registry). Some applications, however, have surprisingly preferred to have a CA for each VO.

The latter decision may be motivated by the complication of deciding which CAs to trust: the documents described above indicate what the CA *intends* or proposes, or claims to do. It would be expensive and impractical to audit the CAs involved in scientific projects for compliance to their policies. However, in accepting such identity certificates, issues of real-world liability arise.

GGF is standardising such policies [BG02] to enable relying parties to use expected vocabulary and levels of trust.

Proxy credentials Certificates that are supplied and signed by Certificate Authorities are typically regarded as *long-term* with, for example, a life span of one year. Even though sites are expected to update the corresponding revocation lists on a regular basis, it was deemed infeasible to do this for every authentication request (as described above). The inherent weakness of this system is indeed the lack of scalable revocation mechanisms (see Section 4.3). As the number of users, resources and applications increases, so does the threat of certificate compromise (through the cracking of keys or by gaining access to poorly protected secret keys). In some applications, where data is not confidential, this risk may be seen a worthy trade-off for efficiency. In others, for example medical groups, the consequences of compromised certificates are very serious.

To address this problem, many grid technologies (including the EU DataGrid) introduce the notion of a second type of certificate, referred to as a *proxy credential*. A proxy credential takes a similar form to that of a standard certificate (for example, X.509 certificate format), but with the distinctive characteristics that it is (i) signed by a grid user and not a Certificate Authority and (ii) granted a short life-time. Users accessing grid resources must generate a proxy credential and use this as part of their authentication process. These credentials must be supported by the user's original long term certificate for the purposes of verifying that they are authentic. By imposing a short life span upon the credential, it is hoped that the risks of compromise are reduced.

The delegation of a user's rights is achieved using short term credentials. A user U delegates rights to another grid entity U' by creating a corresponding short term credential and signing it with U's long term key. U' uses this credential in the authentication procedure (for example, SSL) to access resources on behalf of U; the resources can verify the authenticity of the request by verifying the signature of U supported by U's certificate (i.e. signed by a trustworthy CA to prove the authenticity of U's signature).

This type of delegation is an important requirement within the grid architecture, for example: Components responsible for handling job submissions and finding available resources must be granted certain rights from the users to gather this information; a resource running a job on behalf of some user U may involve accessing other resources on behalf of U; the single sign-on requirement naturally implies the need for delegation.

One of the consequences of running jobs using short term credentials is that it is not always possible to estimate their precise duration and the credential may expire before completion. For the reasons discussed above, it is preferable to keep the life time of a credential short, even if the job is known to require a longer period of time to execute. In this case, it is necessary to deploy a mechanism for handling expired credentials and renew them under certain conditions.

The EU DataGrid have proposed the deployment of a *Proxy Renewal Service* [Dat03] that is responsible for maintaining and renewing expired credentials accordingly. The precise design of this component remains vague in the literature and has the risk of becoming a security weakness in the system. It is vital, for example, to ensure that security checks are carried out before every credential renewal; in other words, one needs to verify the authenticity of the job and user for which the job is being run and verify that the corresponding certificates have not been compromised. Simply renewing a credential without such checks defeats the purpose of introducing short expiry times.

Key management Enabling end users to manage the secret keys associated with their PKI certificates is a significant problem. One of the leading solutions to date is MyProxy

[NTW01], which offers, through a web portal, a repository for key storage and a source for generation of proxy credentials and for job submission. This simplifies the key management hugely, but places considerable requirements on the security — and availability — of the MyProxy server.

4.2 Authorisation mechanisms

Within the grid environment, it is necessary to address the problem of specifying and enforcing local access policies within distributed virtual organisations. Every virtual organisation and resource site will determine and control their local authorisation policies. Furthermore, these may frequently change as users are added and removed, user rights changed according to project requirements and so on. To interoperate on a large scale, it is simply not feasible to have all communities and resource sites interact and update one another every time such a change takes place. There have been various architectural proposals for handling authorisation, for example, Community Authorization Service (CAS) by Globus [Glo] and the Virtual Organisation Membership Service (VOMS) by the EU DataGrid [Dat].

Virtual Organisation Membership Service Within the DataGrid, authorisation has been divided into global and local requirements. The aim is to provide a scalable solution without forcing resources to give up control over their access policies to a centralised server. The global authorisation is handled by a trusted VOMS server, whose responsibility and status is very similar to that of a Certificate Authority for authentication. There is typically one VOMS server per Virtual Organisation. Before submitting a job or resource access request, a user must request an authorisation certificate from, and signed by, the VOMS server. This certificate contains information regarding a user U's membership within the VO and enables resources to determine what access and permissions are to be granted accordingly. The certificate will also contain U's authentication credentials for future authentication purposes.

Separating global and local authorisation mechanisms has the benefit of increased interoperability between sites and their local policies; individual sites have the ability to determine the precise mapping strategy between global authorisation certificates and the resource access rights.

One problem that has arisen in the DataGrid deployment of this mechanism is a lack of consistency when data is replicated. Authorisation was originally bound to the access of physical sites as opposed to data. However, when a user wants to gain access to a particular set of data, a distinction between the authorisation of the logical data and its physical location must be made. One needs to be able to specify that data is accessible to a particular set of users/resources irrespective of its location or replication mechanism.

4.3 Revocation mechanisms

PKI relies upon the periodic distribution of certificate revocation lists (CRLs) in order to allow those relying upon certificates to gain confidence in their present validity. Use of CRLs is patchy; the frequency of their updates (compared to the lifetime of the certificates, and the cost of reliance upon a bad certificate) must be carefully managed. The lack of a scalable revocation mechanism is a serious drawback with this approach.

An alternative is the online certificate status protocol OCSP [MAM+99], wherein one or more servers is established, consolidating CRLs from various CAs and providing a service for checking certificates. This would seem promising for the DataGrid to adopt, but it is not presently envisaged.

When enabling delegation, revocation becomes even more problematic. Many grid projects are keen to allow delegation chaining. As discussed in [BL03], this architecture suffers from many security problems including the lack of revocation when a credential becomes compromised. There is no possibility of immediately revoking access—you have no idea where a particular user's jobs are running. Use of VOMS (Section 4.2) gives the potential to revoke authorisation aside from authentication, but pre-existing authorisation tokens will remain valid for their original lifetime.

4.4 Confidentiality mechanisms

Global Security Infrastructure (GSI) The GSI framework by Globus is deployed within the DataGrid and provides confidentiality of data between sites (using SSL/TLS).

Data on remote resources Most of the current applications within the DataGrid project assume full trustworthiness of the resources upon which data is stored, code is run, results are generated and so on. This extends to trusting the resource to: run the correct code with the correct data and return the actual results; protect all the data against duplication and unauthorised access; adhere to the requirements upon termination (for example, deleting files or executable code). There are no proposed solutions for confidentiality in these cases, thereby limiting the scope of applications.

For the biomedical applications, the data is stored encrypted within the databases to satisfy the confidentiality requirements. When a file is accessed by an authorised user, it remains encrypted while transported from the resource to the user's local site. The user must request the corresponding decryption key separately in order to access the file. Given the stringent confidentiality requirements, it is not sufficient to rely upon encryption alone. The data must be protected while in decrypted form on the local site; the confidentiality mechanism needs to encompass local policies to enforce the necessary level of security.

User and resource privacy The division between the treatment of global versus local authorisation has led to a greater degree of confidentiality for the users and resources. When a user U submits a job request, the component responsible for identifying available resources on U's behalf knows the resources that have agreed access, but does not gain any knowledge regarding reasons why access is denied: U may be on a banned list, but equally the resource may not be available for technical reasons. The status of a user with respect to the local site policies remains confidential.

4.5 Distributed Trust

The issues of trust raised above (in Section 3.5) are not directly addressed by technologies. Conceptions and models of trust have been discussed at length, but little has so far been written for the grid context.

4.6 Integrity, non-repudiation, Freshness

Integrity The GSI framework by Globus is deployed within the DataGrid project and provides integrity of data between sites (through the SSL/TLS protocol).

As discussed in Section 4.4 regarding confidentiality, resources are assumed to be trustworthy. There are no mechanisms in place within the DataGrid project to ensure integrity of the job executions. Thus, when a user receives data that was supposedly generated by a particular source, there is no guarantee that (i) the correct code was run or (ii) the data sent matches the actual data generated on site.

Non-repudiation Non-repudiation is not addressed in the DataGrid project. As grid technology becomes more widely used, non-repudiation will be necessary to provide accountability of resource usage and for accounting purposes.

Freshness Freshness of authentication and authorisation mechanisms is achieved by including a limited time span within the certificates and credentials: they are only valid within the timeframe provided. However, as pointed out by Gong [Gon92], the use of timestamps requires the clocks of the different agents to be loosely synchronised; further, this makes the time-maintenance mechanism everywhere part of the security architecture.

There is no current mechanism for enabling a user to impose some limitation upon the number of times a delegated right may be exercised. This could be captured in the delegation credential, but this would only be effective if the resource was trustworthy and maintained a record of usage.

5 Conclusions

We have seen that grids raise a number of distinctive security requirements. Security is a socio-technical topic, and many of the grid challenges are social in nature. Some of the technical challenges are being met; other security requirements, such as those relating to trust, have much wider implications that need to be considered; others still, have simply not been addressed.

In Section 4, we gave an overview of some of the proposed mechanisms within the EU DataGrid project [Dat]. By breaking the security aspects down in Section 3 and considering how each of them is addressed by the architecture, it becomes rapidly apparent that security within the DataGrid (and many similar projects) is still far from being solved. Where technical solutions exist — such as with PKI — issues such as how to enable users to manage keys effectively remains unclear. Other security weaknesses include: an inherently weak revocation mechanism for the proposed authentication, authorisation and delegation mechanisms, a dependence upon the global synchronisation of clocks and implicit trust assumptions of remote resources to ensure confidentiality and integrity requirements.

Security has proved over the years to be extremely difficult to achieve; millions of pounds are still disappearing every year through Internet fraud. It is therefore not surprising that it is far from satisfactory within projects of scale and complexity as a grid. However, there are a number of contributing factors we have observed in the DataGrid that exacerbate the problem:

Component-centric software development As has frequently been the case in the past, security requirements were not considered during the initial phases of the software development process. This has fractured the overall design and led to a division between the development of security versus non-security components. Due to the shear scale and geographic dispersion of developers and researchers, the project has naturally taken on a component-centric design approach. This has been extended to handle security.

The fundamental problem with this approach is that security cannot be isolated into independent system components; rather it encompasses a multitude of technical and social aspects of the system. To be successfully achieved, security must be factored in from the start and lie at the heart of the overall design. For example, when considering confidentiality, there is no point in solely running SSL between sites. Confidentiality involves the local sites upon which the data is stored and accessed (including the intranet, local machines and users), all grid mechanisms that handle the data (for example, replication mechanisms), as well as the inter-site transport mechanism. The same principle applies to other security solutions.

Composability of security mechanisms It is well known that a system composed of multiple secure components does not necessarily imply that the system as a whole is secure. This problem is magnified within the DataGrid due to the scale, wide range of varying security requirements and its heterogeneous nature. Determining the composibility of the various security requirements and proposed solutions is crucial in assessing the levels of security provided by the system in parts or as a whole. Furthermore, analysing the security of a system ideally requires the use of formal or semi-formal techniques. To feasibly achieve this for a system of this scale, the ability to de-compose the problem is essential. Due to the scale and evolving nature of the DataGrid project, there has been no methodical or consistent approach taken to collate the broad range of security requirements. This exacerbates the difficulty of evaluating the overall security.

Verification of security architecture Experiences of EU DataGrid project has emphasised the need for whole-system security analysis. In many practical software projects (especially of the size and scope of the DataGrid), formal methods do not provide the flexibility, scope and scalability necessary to be of much use. However, depending upon the application requirements, the level of security provided must be evaluated and “proved” with a certain degree of accuracy. Other practical problems that arise include how realistically and formally to capture the breadth of requirements, social policies and trust assumptions. Identifying assumptions upon trust models, user behaviour, and so on need to be incorporated within any security analysis.

Social versus technical requirements Security is not just about technical requirements and solutions; social aspects are a vital part and must be incorporated into the design and deployment. How is this captured within the design? How does one evaluate and enforce non-technical solutions?

Reconciling the old and new The interaction of grid technologies with other security technologies also deserves attention, since here too there are social and technical challenges. The network requirements of grids bring them into sharp conflict with institutional firewalls [Sur02]. These tend to be antithetical to our grid requirements, wherein arbitrary connections may be made to arbitrary hosts; firewalls are typically intended to restrict access to all but a few hosts and services. Firewalls being managed quite separately from the compute resources, it can be very hard to achieve the required degree of access. Bringing the firewall into the grid security domain helps to solve this problem — at a cost of potentially creating a very fragile network of hosts that must trust each other. Message-based firewalls are envisaged in the next generation of grid technology [Glo, Globus 3].

Acknowledgements

Our understanding of this topic has benefited greatly from discussions with members of the UK e-Science Core Programme Security Task Force and the EU DataGrid Security Group. Jon Hillier's practical insights into the operational challenges of firewalls have been very helpful.

References

[ACK+02] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@home: An experiment in public-resource computing. *Comm. ACM*, 45(11):56–61, 2002.

[Aus01] Tom Austin, editor. *PKI: A Wiley Tech Brief.* Wiley, 2001.

[BB03] P. Bourne and K. Baldridge. The new biology and the grid. In Berman et al. [BFH03].

[BFH03] F. Berman, G. Fox, and T. Hey, editors. *Grid Computing: Making the Global Infrastructure a Reality.* Wiley, 2003.

[BG02] R. Butler and T. Genovese. Global grid forum certificate policy model. Memo, 2002. 'work in progress'.

[BGS+03] M. Brady, D. Gavaghan, A. Simpson, M. Parada, and R. Highnam. eDiamond: A grid-enabled federated database of annotated mammograms. In Berman et al. [BFH03].

[BL03] P. J. Broadfoot and G. Lowe. Architectures for secure delegation within grids. Submitted for publication, 2003.

[Dat] *The EU DataGrid Project*. See `http://www.eu-datagrid.org`.

[Dat03] DataGrid Security Co-ordination Group. *Security Design (Deliverable 7.6)*, 2003.

[DZ83] J. D. Day and H. Zimmerman. The OSI reference model. *Proc. of the IEEE*, 71:1334–1340, 1983.

[FK99] I. Foster and C. Kesselman, editors. *The Grid*. Morgan Kaufman, 1999.

[FKT01] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *J. Supercomputer Applications*, 15(3), 2001.

[FKTT98] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pages 83–92, 1998.

[GEO] *The GEODISE Project*. See `http://www.geodise.org`.

[GGF] *Global Grid Forum*. See `http://www.Gridforum.org`.

[Glo] *The Globus Project*. See `http://www.globus.org`.

[GM90] M. Gasser and E. McDermott. An architecture for practical delegation in a distributed system. *IEEE Symposium on Research in Security and Privacy*, pages 20–30, May 1990.

[Gon92] L. Gong. A security risk of depending on synchronized clocks. *ACM Operating Systems Review*, 26(1):49–53, 1992.

[HW03] J. Hodges and T. Watson. Liberty architecture overview. Version 1.1 specification, Liberty Alliance, January 2003.

[Joh03] B. Johnston. Implementing production grids. In Berman et al. [BFH03], chapter 5.

[KWA+01] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky. SETI@home - Massively distributed computing for SETI. *Computing in Science and Engineering*, 3(1):78–83, Jan–Feb 2001.

[Lin97] J. Linn. Generic security service application program interface. RFC 2078, Network Working Group, IETF, January 1997.

[MAM+99] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 internet public key infrastructure online certificate status protocol - OCSP. RFC 2560, IETF Network Working Group, Jun 1999.

[Mic02] Microsoft Corporation. .NET Passport: Balanced Authentication Solutions. White paper, Microsoft, December 2002.

[Net96] Netscape Communications Corporation. *The SSL Protocol Version 3.0*, 1996.

[NT94] B. Clifford Neuman and T. Ts'o. Kerberos: An authentication service for computer network. *IEEE Communications*, 32(9):33–38, 1994.

[NTW01] J. Novotny, S. Tuecke, and V. Welch. An online credential repository for the grid: MyProxy. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing*, 2001.

[RSG+00] Peter Ryan, Steve Schneider, Michael Goldsmith, Gavin Lowe, and Bill Roscoe. *Modelling and Analysis of Security Protocols*. Addison Wesley, 2000.

[Seg00] Ben Segal. Grid computing: The European Data Grid Project. In *IEEE Nuclear Science Symposium and Medical Imaging Conference, Lyon*, 2000.

[SKM+02] D. Stainforth, J. Kettleborough, A. Martin, A. Simpson, R. Gillis, A. Akkas, R. Gault, M. Collins, D. Gavaghan, and M. Allen. Climate*prediction*.net: Design principles for public resource modelling research. In *Proc. 14th IASTED conference on parallel and distributed computing systems*, 2002.

[Sur02] M. Surridge. A rough guide to grid security. Technical report, IT Innovation Centre, 2002. V1.1.