GA–A24990

# SIMPLIFYING FUSIONGRID SECURITY

by
J.R. BURRUSS, T.W. FREDIAN, and M.R. THOMPSON

MAY 2005

GENERAL ATOMICS

## DISCLAIMER

GA–A24990

# SIMPLIFYING FUSIONGRID SECURITY

by
J.R. BURRUSS, T.W. FREDIAN,* and M.R. THOMPSON†

This is a preprint of a paper to be presented at the Challenges of Large Applications in Distributed Environments, Research Triangle Park, North Carolina, July 24–27, 2005 and to be printed in the Proceedings.

*Massachusetts Institute of Technology, Cambridge, Massachusetts.
†Lawrence Berkeley National Laboratory, Berkeley, California.

Work supported by
the U.S. Department of Energy under
DE-FG02-01ER25455, DE-FC02-04ER54698
and DE-AC03-76SF00098

GENERAL ATOMICS PROJECT 30106
MAY 2005

GENERAL ATOMICS

# Simplifying FusionGrid Security

J. R. Burruss[1], T. W. Fredian[2], M. R. Thompson[3]
[1]General Atomics, P.O. Box 85608, San Diego, California, USA
[2]Massachusetts Institute of Technology, Cambridge, Massachusetts
[3]Lawrence Berkeley National Laboratory, Berkeley, California
burruss@fusion.gat.com

## Abstract

*Since inception in 2001, FusionGrid developers have worked to secure computational resources in a multi-institutional environment with geographically dispersed users. Recent improvements to grid security have streamlined the usage and administration of resources. More than simply increasing security, these improvements have made FusionGrid security easier for resource administrators and the fusion scientists that use FusionGrid, allowing them to get work done with minimal inconvenience. Improvements in authentication, authorization, and data handling have been welcomed by fusion scientists and promise to ease the burden of adding new resources to the grid.*

## Introduction

The National Fusion Collaboratory [1][2] is a virtual organization (VO) consisting of researchers from the three major U.S. plasma physics research centers and collaborators from other universities and labs. The collaboratory has created a computational grid, FusionGrid, consisting of computational services and data repositories. FusionGrid services are geographically dispersed and are used by fusion scientists world-wide.

The purpose of FusionGrid is to provide new capabilities to fusion scientists in order to advance fusion research. One thrust of work is to simplify the use of large, complex fusion codes, many of which are decades old and written in old versions of Fortran. FusionGrid also aims to unify data access, simplify administration, and streamline security processes to enable researchers from different organizations to work together in a single virtual organization. Other research thrusts include the development of collaborative tools for geographically dispersed researchers, and the development of a collaborative tokamak control room.

This paper focuses on the impact of cyber security on FusionGrid, and how security has been made simpler to the benefit of fusion scientists, developers, and system administrators. Careful selection and revision of technologies, as well as the development of a new authorization system, have led to simpler security and more productive scientists.

## Security Requirements

A defining characteristic of a grid is the desire of multiple sites to make resources available to users from administratively and geographically distributed organizations. This imposes a need for users to have a *grid-wide identity* and a means by which they can authenticate themselves as that entity at each site. It is also highly desirable if the user can authenticate using a passphrase once, referred to as *single sign-on*, and have authentications at other sites derived automatically from some limited lifetime token granted by the original sign-on.

One of the common usage scenarios on the FusionGrid is to start a compute job at a remote site, which will then contact a third site to read or write data. The compute job operating on behalf of the user needs to be able to authenticate and act as a proxy for the user at such a third site. This is referred to as having the ability to *delegate rights* to processes acting on your behalf.

Another requirement of the FusionGrid is to enable *multiple stakeholders* for a single resource set access policy for that resource. These stakeholders may be remote from each other and from the resource they control. Some means for them to easily see and set resource access is needed.

A general requirement of all access systems is to adhere to the *principle of least-privilege*. A user should only be granted the rights that are needed to accomplish his task. This is especially important for grid resources that are allowing access to users from a number of other domains.

The most important requirement is one that is often neglected in discussions of computer security: FusionGrid security must be usable by the fusion scientists that are the ultimate users of FusionGrid. Any system that is too complicated for the scientists to use is unacceptable.

## Authentication

For a virtual organization like FusionGrid, authentication becomes a challenging problem because of the different organizations involved. Scientists from the different fusion research facilities and universities must somehow work together and be made identifiable on FusionGrid. Here, the traditional approach of assigning usernames breaks down because, with the different local organizations, naming conflicts quickly arise between a scientist's username at one institution and their username at another institution. Furthermore, traditional login brings with it the onus of logging in to each separate resource in the virtual organization. This is problematic because, as any systems administrator can testify, scientists will frequently forget their username and password to machines which they log in to only infrequently. For these reasons, some security system had to be developed to uniquely identify users and provide them with way to log in to the entire FusionGrid as opposed to each individual resource.

Early FusionGrid investigations identified the use of X.509 [3] credentials and the Globus Security Infrastructure (GSI) [4] as mechanisms to uniquely identify users in a virtual organization and to provide a single sign-on capability. A DOEGrids certification authority (CA) was used, and X.509 certificates were issued to FusionGrid users. A Web interface was provided for fusion scientists to obtain their X.509 certificates. Each user managed their own X.509 credentials, keeping their certificate and private key files in their home directories and Web browsers. In this way, each FusionGrid user could be uniquely identified by their certificate. GSI enabled a powerful single sign-on capability: users could, using their DOEGrids credentials, sign on to FusionGrid a single time without needing to log in to each resource used.

This system was a success, and provided fusion scientists with the capability to get their work done in a virtual organization. However, scientists had a very hard time managing their own certificates. The process of exporting and importing credentials from their Web browsers, converting the credentials to the required formats, and installing their credentials in the appropriate place with the correct file permissions in their various home directories was a source of endless frustration to scientists. Installing in one home directory on one computer is merely an inconvenience, but it is completely unmanageable to do so for multiple accounts at multiple sites, especially while on travel to other fusion experiments across the world. To make things worse, this frustrating process of obtaining and installing their certificate was, for each scientist, their first impression of FusionGrid. If just *signing up* for FusionGrid was this difficult, what could they expect from actually *using* FusionGrid?

Once they finally obtained and installed their certificates — almost always with lots of help from FusionGrid programmers and administrators — scientists had more certificate problems to look forward to: the 1-year lifespan of certificates meant that the process had to be repeated every year. After a year's time it was generally true that the scientists would not only forget how to manipulate their certificates, but would forget which Web browser on which computer was used, or would have uninstalled their original Web browser. Furthermore, renewed certificates were not valid until the original certificate expired; this meant that a user could not renew their certificate early, because the new certificate would be invalid. If they did replace their original certificate with a new one that was not yet valid, it was time to contact their local system administrator for a restore from backup. The workaround to this replacement problem was to renew the certificate, but keep it in a standby area until the previous certificate expired. However, on at least one occasion this meant that a scientist had to avoid starting a long-running job the night before his certificate expired—this was not just an inconvenience, but an impediment to getting work done.

FusionGrid developers have attacked this problem on several fronts. First, developers worked on better training for scientists to allow them to get comfortable with the idea of certificate management. To this end tutorials have been presented at the American Physical Society Division of Plasma Physics meetings [5] and FusionGrid Web pages have been improved.

Second, FusionGrid developers created an alternative to the DOEGrids CA browser interface. A set of command-line scripts was written to allow users to request and replace their certificates without using a browser [6]. These scripts hide some of the DOEGrids CA interface generality and place the certificate and private key in the location where the Globus software expects them. Additionally, the replace feature addresses the problem where renewed certificates were being installed before becoming valid by instead

replacing the original certificate with a new certificate that is already valid.

Ironically, developers started with a Web approach because it was thought that a Web interface for certificate management would be easier for users than a command-line interface such as the one provided in the Globus Toolkit™. In practice, there are two main problems with the browser approach. First, browser certificate management interfaces are not uniform across different browsers and have not always been correct in their handling of certificates. Second, the user ultimately needs the certificate and private key stored as a local file for use by the job submission client rather than in the internal storage of their Web browser.

Although user education and improved interfaces were beneficial, it was the third approach to solving the certificate management issue that ultimately had the most impact. Developers decided, since certificate management was such as hassle, to altogether remove the burden of certificate management from scientists. FusionGrid developers chose to adopt MyProxy, [7] a system for storing and retrieving credentials. Under MyProxy, users keep their credentials on a remote server. When credentials are needed, a single command is typed to retrieve the credentials and create a new limited-lifespan delegation of the credentials. Thus, users do not need to copy their credential files around and install them on each client machine. It was decided that, the technical details being unimportant to the scientists, MyProxy was to be presented to them as the way to "log in to FusionGrid."

FusionGrid developers created a *credential manager* which consists of a MyProxy server, an SSL-enabled Apache Web server, and a custom Web interface, all of which run on a dedicated and secured host. The Web interface allows scientists to request a FusionGrid credential by typing in some information about themselves including a user name and password that they will use for their grid login.

Figure 1 illustrates the registration process. When the user first registers (1) with the credential manager it generates a new X.509 certificate and private key. The credential manager asks the CA to sign the certificate (2) which the CA does after getting the request approved by one of the FusionGrid registration agents, i.e. a human with the authority to approve certificate requests. Then the credential is loaded into the MyProxy server (3) and the username and distinguished name are sent to the ROAM authorization database (4) discussed in the next section.

It was necessary to commission a new CA because the original DOEGrids CA policy does not allow third party storage of private keys, something that is absolutely required for centralized credential management, but could diminish the non-repudiation value of the private keys.

The use of the FusionGrid credential manager was a tremendous success. Without exception, scientists indicated that they approved of this simpler approach. To make the transition easier, users with old certificates can obtain new MyProxy-enabled certificates from the new FusionGrid CA and have those certificates linked in the authorization database discussed below. Because the certificates are linked, users can use either set of credentials and inherit the same set of permissions.

## Authorization

FusionGrid authorization requirements are straightforward enough: resource stakeholders need to be able to control who has access to their resources. The author controlling access to his or her code, the system administrator controlling access to his or her computer, and the site security administrator controlling access to his or her site: each of these stakeholders must be empowered to turn on and off access as required.

The first technology used for FusionGrid authorization was the "grid-mapfile" functionality of the Globus Resource Allocation Manager (GRAM) [8]. Essentially, a grid mapfile is a text file that maps user certificates to local user accounts. This functionality is standard with GRAM and requires no extra software to use, just a text editor for working with the grid mapfile.

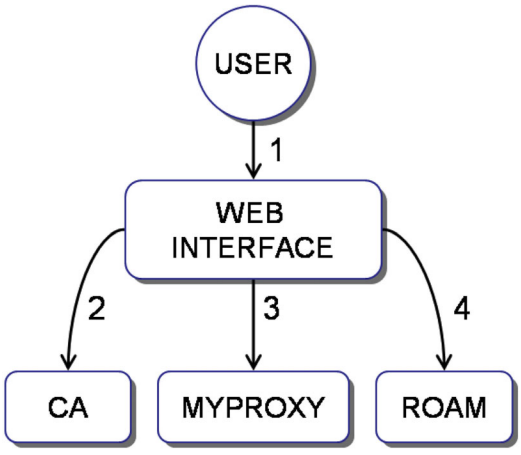


**Fig. 1. Users register with the credential manager through a Web interface.**

Early use of grid mapfiles showed that authorization was difficult because of the need for each individual host on FusionGrid to maintain a separate grid-

mapfile to map FusionGrid certificates to local accounts. Having mapped certificates to local accounts, it was left to each individual resource administrator to implement both grid-wide and local authorization policies. Experience demonstrated that it was hard to maintain coherence with mapfiles distributed across multiple machines at multiple sites. Furthermore, the use of text mapfiles precluded variable account mapping because of the lack of wildcard specification and the need to edit each individual mapfile by hand. This is especially problematic when the same user is mapped to different local accounts on one machine depending on the resource being used, or when a resource is spread across several machines. What was needed was a flexible system to provide grid-wide authorization and account mapping.

Initial efforts to improve authorization in FusionGrid focused on the Akenti authorization system [9]. Akenti is an authorization system designed to handle distributed resources controlled by multiple-stakeholders. It was originally used within FusionGrid, but its implementation of authorization information as distributed, digitally signed documents made some of the desired operations difficult. In particular it is not easy to list all the users and resources of the VO, or to check on all the outstanding authorizations for a given resource.

FusionGrid developers also looked at the Community Authorization Server (CAS) [10], the Virtual Organization Membership Service (VOMS) [11], and Shibboleth [12]. (See the section on related work for a brief description of these systems.) However, each of these systems implement the *push* model of authorization, where the user must first contact an authorization server to get some credentials and then present them to the resource provider. These credentials are in addition to the X.509 credential used to establish identity and may consist of attributes such as roles and account id or grants of specific permissions. The provider must then verify the credentials and often do additional local authorization checks. This was a problem because FusionGrid architects wanted to keep the authorization path as simple as possible, both for scientists and for developers. Furthermore, each of these designs focuses on defining users and their attributes or permissions; they do not attempt to define the resources of a VO, something that was desired for FusionGrid. Any system that would empower stakeholders to control access to their resources must first define those resources.

Eventually it was decided to develop a new authorization system, one that would meet the needs of resource stakeholders while being as simple as possible and easy for scientists to use. The result was the Resource Oriented Authorization Manager (ROAM) [13].

The ROAM information model consists of a framework of *resources*, *permissions*, *users*, and *authorizations*. Everything in the ROAM universe is one of these four types of things. A resource is typically a grid service, but it can also be an entire site, like the Alcator C-MOD or DIII-D fusion experiments. A user is any uniquely identified consumer of resources. When the FusionGrid credential manager enters a new credential in the MyProxy server, it enters the user name, the distinguished name and other user information into ROAM. Thus, ROAM always has the list of all authorized FusionGrid users.

A permission is a type of usage for a resource; in other words, a permission is a way in which a resource is used, for example “read” and “write” for a database, “access” for a site, or “execute” for a code. An authorization is a grant of a specific permission for a particular user on a specified resource. The key to the ROAM model is this: clearly indicate resources, define stakeholders for those resources, and empower stakeholders to control access to their resources.

ROAM authorization information is stored in a database, and a Web front-end is used to view and change that authorization information. The choice of a Web front-end is important; it is easier for all involved to point their browsers to a URL than to either edit text files or use special custom GUIs.

ROAM avoids the push model of authorization. Instead, clients connect to resources as they would normally, using an X.509 proxy credential from MyProxy to authenticate (Fig. 2). The resource then consults ROAM to see if the connecting user is authorized. In this way the authorization path is completely transparent to the user.

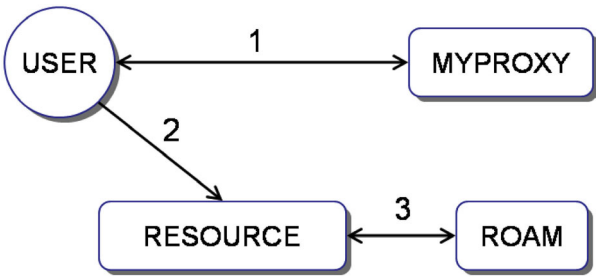


**Fig. 2. A users first signs on to FusionGrid using MyProxy, then accesses a resource directly; the resource then checks ROAM for authorization.**

The recent adoption of ROAM is having a positive impact. ROAM is currently being used for authorization management for the GATO [14] FusionGrid service. It is also being used for access control to MDSplus [15] datasets at DIII-D and Alcator C-MOD from remote clients (which connect to secure MDSplus, discussed in the next section of this paper).

Initial feedback from FusionGrid users and engineers indicates that the ROAM Web interface is very convenient. The simple Web interface for managing permissions is less prone to errors than editing configuration files. It is never necessary to type in the full distinguished name from a certificate when authorizing a user as the Web interface will provide a list of users from which to choose. It is easy for a resource administrator to grant the "Admin" permission to additional users so that they can help administer a grid service. A user can easily request the permission to use a resource or make an inquiry regarding a particular resource using the Web interface. All user requests for permission to use a resource as well as permission grants or revocations done through the Web interface will generate an email notification to the resource administrators for that particular resource. This built-in email feature creates an informal audit trail.

The ability to define an entire site as a resource has allowed the site administrators to feel confident that they can quickly remove all access to their sites in case of an emergency, and easily audit all users of their site.

## Secure Data Storage and Transfer

In the magnetic fusion research community there is the need for remote collaboration and the rapid exchange of data. This data may be the measurements taken during the operation of fusion experiments or data generated by modeling codes. Currently most fusion experiments are pulsed experiments, with pulses occurring at a rate of once every 15 to 30 minutes with each experiment generating from hundreds of megabytes up to a few gigabytes of measurements per pulse. This data set consists of several thousands of different measurements and each scientist may be interested in a subset of these measurements. The data is analyzed and visualized immediately after the experiment occurs. Since the entire data set generated for an experiment is so large, and since it is necessary for remote collaborators to view a subset of the data and perhaps write analysis results back to this data set, using simple file transfers is not practical.

To meet these needs for remote data access the MDSplus data system (used by most fusion experiments for data handling) was extended nearly ten years ago to allow remote clients to read and write data to the main data servers. This extension used a simple protocol layered on TCP for data exchange. The security model for this data exchange was based on a simple user mapping similar to the original Berkeley rhosts mechanism. The client sends the name of the remote user and the server then uses the username and the IP address of the connecting client to look up a mapping to a local account. A text file is used for this mapping of connecting clients to local accounts, much like the grid mapfile in GSI.

While it is true that this host-based authentication mechanism works, there are two problems. First, it is obviously quite insecure as it is easy to present a different username and to spoof the IP address. Second, the use of host-based authentication means that, for users that work from dynamic IP addresses (e.g. on their laptops in a coffee shop), a proxy with a known IP address is required; this is at best an inconvenience and at worst another point of failure and performance bottleneck for data transfer. The increasing amount of remote collaboration coupled with increasing Internet security concerns has made it necessary to explore more flexible and secure solutions for the authorization of remote data access.

After exploring various solutions such as the use of OpenSSL or Globus Grid Security Infrastructure (GSI) it was determined that a secure version of MDSplus could be created using GSI. GSI provided valuable features such as single sign-on and credential delegation. A client using GSI can authenticate to a remote server using an X.509 certificate. It is the certificate—not the originating host—that uniquely identifies the user. Additionally, the delegation feature means that the remote server can then connect to other servers on behalf of the originating client, and that the subsequent authorization is based on the client credentials. This was a good match for our security requirements. It was decided to modify the existing MDSplus client/server software to layer on top of the GSI instead of TCP.

The modifications were successful, and MDSplus can now use the GSI for communication between clients and servers. Furthermore, because of mutual authentication, not only can the server securely authenticate with the client, but the client can now be certain that he is communicating with the desired server. All communication packets are digitally signed, so any packet tampering is detected.

Additionally, secure MDSplus can be configured to call the ROAM authorization system, both for authorization and for account mapping. For secure MDSplus,

the context information in ROAM is being used for group mapping: clients are mapped to specific groups based on the optional context information, thus enabling fine-grained access control. For example, files can be made group-writable, limiting remote write access to those clients mapped to the appropriate group.

The new GSI-secured MDSplus is being used with FusionGrid computational services with success. Secure MDSplus is also being used by offsite collaborators for more general data access purposes. It works with traditional X.509 certificates as well as delegated proxies of the sort used by MyProxy. The impact has been positive and has allowed collaborators to securely and reliably access their data.

## Related Work

Several other other grids have addressed some of the same issues. The European Data Grid developed the Virtual Organization Membership Service (VOMS) [11] in order to manage user information including attributes such as groups, roles and capabilities that a user might have. A user who wishes to use a resource first contacts VOMS to get a signed copy of its privileges, in this case an attribute certificate [16] that can be presented to the resource provider. The resource provider will then contact a local authorization server to evaluate the privileges and possibly grant access to the user. VOMS emphasizes the assignment of groups and roles by the VO and mostly leaves the granting of access based on those attributes to the resource provider.

The Earth System Grid (ESG) [17] uses the Globus Community Authorization Service (CAS) [10] to implement a central authorization policy repository. CAS stores access information for groups or classes of users with respect to any grid resource. The user contacts the CAS server to get a proxy certificate [3] that contains access rights in a delegated-rights extension. The user then uses that certificate to authenticate to a resource provider. The resource provider may in turn call a local authorization server to verify that the CAS had the authority to delegate the rights and to do any further fine-grained access. ESG provides a tool for analyzing data that relies on FTP to fetch data from various sites. GridFTP was modified to look at the access rights carried in a CAS proxy and use them to grant read or write access to data. This allows a site to grant access once to the CAS server after which the data owners can grant access to specific files via the CAS policy repository.

ESG has also recently integrated a MyProxy server with their portal, [18] allowing the same username/password login that the FusionGrid Certificate Manager allows. In their case they chose to run a local CA; FusionGrid developers decided to leverage the ESNET CA infrastructure. At this point it is too early to say whether it would have been better for FusionGrid to manage its own CA as opposed to outsourcing this task. This decision was motivated by the desire to put this critical piece of security infrastructure under the administration of a professional organization and to make FusionGrid certificates as acceptable as possible to other virtual organizations.

## Future Work

For scientists in magnetic fusion research, Mac OS X and MS Windows are the most common platforms for desktop and laptop machines. However, currently GSI, and thus secure MDSplus and the MyProxy client, do not run on these platforms. Having the FusionGrid client tools conveniently available to the scientists is a primary goal of the FusionGrid. For this reason, porting enough of GSI to support these client-side tools is a high priority task. One strategy under consideration is to identify those portions of GSI that are needed for secure MDSplus and the MyProxy client code "myproxy-get-delegation", then build a "GSI-lite" library with just those pieces. This reduced library would be easier to port to OS X and Windows. After porting the GSI-lite library, FusionGrid developers could then build secure MDSplus and myproxy-get-delegation against this separate ported code base, thus giving scientists the ability to use client tools on their preferred platforms. Since secure MDSplus could be used for job management, there would be no need to port GRAM. This task of building and porting GSI-lite is not an ideal solution, not just because of the work involved to port the code, but also because of the increased maintenance work needed to maintain the extra code and the risk of creating software which diverges from mainline GSI. However, with the Globus software development moving onto GT4, the probability of a complete port of GT2 to Windows and OS X seems unlikely.

This problem of platform compatibility has increased interest in research into the use of Web portals in fusion science [19]. One of the chief benefits of Web portals is that they can free the user from the need to use special client code, and enable them to stick to a simple, well-understood client application: the Web browser. Recent work was done to create a Web portal [20] for launching jobs for the TRANSP [21] service on FusionGrid. Although the prototype portal was able to launch jobs, it was not convenient because

certificates from the original DOEGrids CA had to be manually copied and installed on the portal server. Now that FusionGrid is using MyProxy, more work is being done to look into the portal approach, but with the use of MyProxy instead of self-managed certificates. It has already been demonstrated that a user can log into a portal using their MyProxy username and password, and have the portal retrieve a delegated credential, then act on the users behalf using that credential. FusionGrid already has a small and simple Web page that can, using server-side scripting, obtain user credentials from MyProxy and check user authorizations or run test jobs on behalf of the user. Indeed, the ROAM Web interface uses similar server-side scripting to retrieve credentials in those cases where the user does not already have their credentials stored in their browser.

Another direction to enhance FusionGrid security would be closer integration between ROAM and the FusionGrid Monitoring (FGM) [22] system. FGM provides a central logging facility for all FusionGrid jobs. The log files are accessible to everyone through a Web interface and are used by users to see the status of their long running jobs. Currently ROAM provides a secure Web interface and keeps a log of all authorization queries, allowing a stakeholder to see both successful and failed attempts to use a resource. If the two servers were to work together, the authorization attempts could be posted from ROAM to FGM, where the users could see them, and the amount of audited computing resources used could be sent from FGM to ROAM where the information could be kept securely and be available to the stakeholders. To illustrate, if usage of a particular resource involved CPU time, then FGM, which is being informed of job start and end times, could post the usage time to ROAM which would keep a history of CPU time usage by each user. In addition, MDSplus could inform ROAM of user data storage usage.

With detailed auditing information stored in ROAM, access policy could be extended to add a quota capability. Resource administrators could impose quotas, then save that information to ROAM. Authorization queries would then be structured not just to confirm that a user is authorized, but also confirm that they have not exceeded their quotas for the specified resource. Such a system would empower resource stakeholders with a convenient way to enforce usage quotas for their resources.

Should resource stakeholders determine that their authorization information is not to be made public, authorizations could be marked as private in the ROAM database, preventing authorization information from being posted to the FGM. The chief benefits of a closer integration are standardization of messages (useful for scientists that would prefer to learn a few standard messages as opposed to multiple custom messages) and removal of duplication of effort, making life easier for FusionGrid resource developers and leading to more rapid FusionGrid service development.

Since FusionGrid has come to rely on the MyProxy server and ROAM during data access and job submission, they need to be always available. Work will be done to start running a backup version of each server at a different site in the near future. Since both systems keep their data in a compact form and the data does not change too rapidly a simple copying of data to the backup server once a day should be sufficient for FusionGrid needs. If the primary server or site is down, the client interfaces can switch over to the backup. The FusionGrid implementation of ROAM uses an object-relational database that supports replication, so perhaps this built-in functionality could be leveraged.

## Conclusion

The work done to improve FusionGrid security has been a success. Ask a typical scientist, and they will tell you that security is just something that makes it harder for them to get their work done. In this case, however, FusionGrid security was enhanced not just to increase security, but also to make security simpler for the fusion scientists that use FusionGrid to get work done. ROAM is not any more robust than alternative authorization systems, but it is easier to use both for administrators and users. GSI-secured MDSplus is certainly more secure than regular MDSplus, but perhaps as importantly, the migration to certificate-based authentication also makes it easier for scientists to work from home or while on travel. The switch to centrally stored credentials has made it much easier for scientists to use their credentials. It can also be argued that credentials are safer when stored on a secure server than on multiple private workstations that may be compromised without the owner's knowledge. These advances have made FusionGrid friendlier to users without sacrificing security.

## Acknowledgments

This work was funded by the SciDAC project, U.S. Department of Energy under contract number DE-FG02-01ER25455 and cooperative agreement number DE-FC02-04ER54698, and by the Director, Office of Science, Office of Advanced Science, Mathematical, Information and Computation Sciences of the U.S. Department of Energy under contract number DE-AC03-76SF00098.

## References

[1] J.R. Burruss et al., "Remote Computing using the National Fusion Grid", *Fusion Engin. and Design* **71**, 251 (2004).

[2] D.P. Schissel, "Building the U.S. National Fusion Grid: The National Fusion Collaboratory Project", to be published in *Fusion Sci. and Technol.* (2005).

[3] R. Housley et al., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, http://www.ietf.org/rfc/rfc3280.txt (2002).

[4] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids", *Proc. 5th ACM Conf. on Computer and Communications Security,* pp. 83-92 (1998).

[5] D.P. Schissel, "Grid Computing and Collaboration Technology in Support of Fusion Energy Sciences", to be published in *Phys. Plasmas* (2005).

[6] Scripts for Certificate Management http://dsd.lbl.gov/FusionGrid/CertScripts.html

[7] J. Novotny et al., "An Online Credential Repository for the Grid: MyProxy", *Proc. 10th Intl. Symp. on High Performance Distributed Computing*, IEEE Press (2001).

[8] K. Czajkowski et al., "A Resource Management Architecture for Metacomputing Systems", *Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing* **62** (1998).

[9] M. Thompson, A. Essiari, S. Mudumbai, "Certificate-based Authorization Policy in a PKI Environment," *ACM Transactions on Information and System Security (TISSEC)* **6**, 4, pp: 566-588 (2003).

[10] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration", *Proc. IEEE 3rd Intl. Workshop on Policies for Distributed Systems and Networks* (2002).

[11] R. Alfieri , R. Cecchini, V. Ciaschini, L. dell 'Agnello, A. Frohner, A. Gianoli, K.L Lorentey, and F. Spataro, "VOMS: an Authorization System for Virtual Organizations", presented at the *1st European Across Grids Conf.*, Santiago de Compostela, February 13-14, 2003.

[12] S. Cantor, ed. "Shibboleth Architecture: Protocols and Profiles", Working Draft 05, 23 November 2004 http://shibboleth.internet2.edu/, accessed January 24, 2005.

[13] https://roam.fusiongrid.org, accessed February 3, 2005.

[14] L.C. Bernard, F.J. Helton, R.W. Moore, *Comput. Phys. Commun.* **21**, 37 (1981).

[15] T.W. Fredian, J.A. Stillerman, "MDSplus: Current Developments and Future Directions", *Fusion Engin. and Design* **60**, 229 (2002).

[16] S. Farrel and R. Housley, "An Internet Attribute Certificate Profile for Authorization", RFC3281 (2002).

[17] Earth System Grid, http://www.earthsystemgrid.org/, accessed February 7, 2005.

[18] PURSE: Portal-based User Registration System; http://www-unix.grids-center.org/r6/ecosystem/security/purse.php, accessed February 7, 2005.

[19] M. Thomas, J. Boisseau, M. Dahan, E. Roberts, A. Seth, T. Urban, D. Walling. "Experiences in Developing a Web Services/OGSI Grid Portals Toolkit," submitted to *Concurrency and Computation: Practice and Experience*, 2005.

[20] GridPort, http://gridport.net/

[21] http://w3.pppl.gov/transp, accessed February 3, 2005.

[22] S. Flanagan et al., "A General Purpose Data Analysis Monitoring System with Case Studies from the National FusionGrid and the DIII–D MDSplus Between Pulse Analysis System", *Fusion Engin. and Design* **71**, 263 (2004).