Deploying the Globus Security Infrastructure in a Production Environment: Testing and Evaluation

M. Draoli

Servizio Reti di Telecomunicazioni del C.N.R., Roma, Italy

C. Gaibisso, D. Giannelli

Netlab, Istituto di Analisi dei Sistemi ed Informatica del CNR, Roma, Italy

Abstract

The Globus Toolkit emerged among several projects as the standard de facto for the design of an infrastructure for computational grids. The Globus Security Infrastructure (GSI) provides security features that integrate and extend standard protocols for distributed systems with original solutions. In this paper we investigate the functional correctness and effectiveness of the GSI features with respect to the main security services peculiar of a production environment. With this aim, we design and deploy a multiplatform, multiversion and multisite testbed for a computational grid. We then define a formal plan of tests and accomplish it in our testbed. Our results show that: message integrity, authentication and non repudiation are well addressed; access control and availability are problematic; message confidentiality was not implemented in the software release available at the time experiments have been accomplished. These results point out that GSI can be transferred to a production environment only if supported with a series of countermeasures aimed to reduce risks implied from a not satisfactory user credentials management and a lack of an effective monitoring system. Finally, we discuss the main points to be fixed in the deployment of a computational grid, such as the integration with Certification Authorities other than the one provided by Globus, and the adopted countermeasures mainly consisting in some additional features, such as an automatic tool for grid user management, a tool for advanced local access control, and a monitoring system for grid resources.

1. INTRODUCTION

The term grid was coined in the mid 90s to denote a proposal for a loosely coupled infrastructure, supporting a wide range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering, such as distributed supercomputing or teleimmersion [1]. The real and specific problem underlying the grid concept is to coordinate the sharing of computing power, storage, data, equipment and other resources in dynamic, multi-institutional virtual organizations, intended as a coordinated set of individuals and institutions together with some sharing rules defining what is shared, who is allowed to share and under which conditions. There is a general consensus in literature regarding the fundamental characterizing principles of a computational grid [2]. These principles are: heterogeneity, i.e., a grid should gather multiple different resources and span several administrative domains, distributed in distinct geographical areas; scalability, a grid should scale from few resources to millions without a considerable performance degradation; dynamism and adaptability, grid applications should not be sensible to mutable conditions of resources availability. Current distributed computing technologies do not address these concerns. In this sense, grid applications considerably expand distributed systems paradigms, with a particular focus on security and allocation, indexing and access management of resources. As a consequence, despite of the great research and experimental activity of the last years [3], most of the issues regarding the design and the deployment of a grid infrastructure are still object of investigation, among which: the integration of software and hardware components in an environment with coordinated network resources; the implementation of a middleware providing transparency and usability of grid components; the development of management and auditing tools for the software grid infrastructure; the development and optimization of distributed applications taking advantages from the grid technologies. During the past years, several projects addressed these issues, such as Legion [4], Netsolve [5] and Unicore [6]. Yet, all these

initiatives rose from a need to support existent applications or defined problems. That is why, thanks to its flexibility, the Globus Toolkit [7], or Globus in what follows, emerged as the standard de facto middleware for grid infrastructures. Globus provides a set of basic integrated grid services, organized in software modules, for security, resource allocation and management, communication, information, monitoring and access to remote data. Table 1 lists the basic modules together with their names, adopted acronyms and a brief description of the provided services.

Addressed Issue	Module Name	Acronym	Module Description
Security	Globus Security Infrastructure	GSI	Authentication and related services
Information Infrastructure	Meta Directory Service	MDS	Distributed access to information about state and structure of resources
Resource Management	Globus Resource Allocation	GRAM	Resource allocation and process management
Data Management	Globus Access to Secondary Storage	GASS	Remote access to data via network interfaces
Communication	Nexus	-	Unicast and multicast services
Health and Status	Hearth Beat Monitor	HBM	System and resources monitoring

TABLE 1: Basic Globus Modules

In this paper we focus on security issues concerning computational grids deployed with Globus. The aim is evaluating the GSI model (release 1.1.3) with respect to its usability and functional correctness in a production environment. To reach our goal, we first define the main security services peculiar of a production environment for computational grids, by investigating on currently available security standards: authentication, non repudiation, access control, availability, and communication integrity and confidentiality. Then we design and deploy a multiplatform, multiversion and multisite testbed for a computational grid. While deploying the testbed, several real world problems have been encountered, such as those related to the dynamic usage of communication ports and, more generally, to the best practice for firewalls configuration. We face and solve some of them and discuss the main points to be fixed in the deployment of a computational grid. Finally, we define a formal plan of tests, aimed to verify the functional correctness and usability of GSI security features and accomplish it in our testbed. Our tests make it possible to evaluate the risks implied by the adoption of GSI in a production environment. Such risks derives from a not adequate grid user management on local resources, the low level protection offered by the Globus Certification Authority and the lack of a service availability monitor. We then propose some additional services and countermeasures to reduce these risks: the integration with Certification Authorities other than the one provided by Globus, an automatic tool for grid user management, a tool for advanced local access control and a flexible grid monitoring system. Furthermore, some guidelines are given to effectively build these services. We conclude that GSI can be transferred to a production environment if supported by and integrated with these countermeasures. The paper is organized as follows: in section 2 the main services distinctive of a production quality security system for computational grids are identified and some insight on the main features offered by GSI is given. Section 3 illustrates the project requirements and activities aimed to the deployment of our testbed. Section 4 deals with the formal plan of tests we designed, its implementation and the obtained results. Section 5, identifies a series of services to be integrated with or to be added to those offered by Globus and discuss the main points of their realization and integration. Section 6 contains some concluding remarks and considerations.

2. GSI SECURITY FEATURES

Security is a crucial issue in modern systems, and it gained more and more relevance in reference to the increased systems connectivity and the rapid diffusion of the Internet and web technologies. Among several characterizations proposed as standards for computer security, we rely on the IT security recommendations given by the European Community [8]. According to this document, the fundamental services in defining a security policy are:

1. Authentication: attests the identity of parties or messages.
2. Access Control: regulates accesses to resources from subjects or from other resources. Each resource has assigned a set of privileges defining from who it can be accessed and under which circumstances.
3. Confidentiality: protects a communication against passive attacks, that is disclosure of sensible information. Integrity: protects a communication against active attacks, that is tampering of sensible information.
4. Availability: provides assurances with respect to the availability of data and services.
5. Non Repudiation: prevents subjects from denying the fatherhood of their actions.

GSI provides basic security functions that can be used by other modules as well as integrated and extended in application software. Determining where security must be addressed in an infrastructure is a challenging problem. As an example, each layer of TCP/IP can be hardened with security features [9]. The Globus approach is considerably different. Actually, the toolkit is not organized in horizontal layers, but rather in integrated vertical modules, addressing different needs: secure [11] communication among grid elements; support for inter-organizational administrative policies, so as to avoid the need of a centralized management of security systems; support for single sign-on [12], i.e. users and applications must only log on once per grid session; support for credentials delegation [13], so as to let applications run autonomously new applications, whenever needed; integration with local security systems [14], so as to inhibit grid environment to interfere with local policies. With respect to the above listed fundamental security services, GSI, which is compliant to the Generic Security Service API [10] standards, provides the following features:

1. Authentication: requires mutual authentication of every party joining a session. The authentication is granted by a public key infrastructure: each entity in a grid (a user, a host, an application) owns a X.509v3 [15] compliant certificate, trusted by some certification authority. At the very beginning of a communication, an initial handshake takes place with the exchange of the respective certificates. In order to grant a secure exchange, GSI relies on the services offered by OpenSSL communication libraries [16] to exchange certificates.
2. Access Control: is accomplished through the usage of plain text files named grid-mapfiles, containing a mapping between certificates subject names and local accounts. Each local system in the grid has its own grid-mapfile, and a remote user is allowed to access resources if and only if the local system grid-mapfile maps the subject name to a local id.
3. Confidentiality: is not provided. Though OpenSSL supports communication confidentiality, legal constraints imposed by USA didn't allow to provide communication confidentiality features in the current release.
4. Integrity: relies on OpenSSL libraries, that provide communication integrity features implemented by default in Globus.
5. Availability: is often not considered in literature, when it comes to a model design. Nevertheless, in a production environment we cannot expect user not having assurances regarding the availability of what they pay for. GSI provides mechanisms to grant availability of data owned by a user on a remote resource. These are achieved by means of secure communication protocols, such as https [21]. As far as services availability is concerned, Globus relies on a dedicated module, the HBM, that manage a limited set of grid events [17].
6. Non Repudiation: is crucial in a production environment, as no feasible accounting model can be offered if grid user can deny their actions; as long as a public key infrastructure is used, identities are attested by the Certification Authority. As a consequence, parties involved in a communication must mutually trust the respective Authorities. GSI also provides logging features in order to track grid users actions.

3. TESTBED DESIGN AND DEPLOYMENT

We designed and deployed a multiversion, multiplatform and multisite testbed, according to the fundamental principles characterizing production computational grids. This testbed consists of several LAN segments, located in two different sites connected by geographical links.

Some of the segments are administrated inside the same site, referred to as Headquarter in what follows, by the Communication Network Department (SRT) and the Data Processing Centre (CED) of the Italian National Research Council (CNR). The resources located in these segments are directly accessible from the Internet. The other site is operated by Netlab, the multimedia and network laboratory of the CNR Institute of Systems Analysis

and Informatics. The site is protected by a firewall with its own Network Address Translation service (NAT), and has a Domain Name Service (DNS) schema *not aligned* with the external one, i.e. the resource is identified by different DNS names inside or outside the firewall.

This configuration is particularly challenging with respect to our interests, as GSI relies on direct and reverse DNS queries to grant mutual authentication for communications among grid parties.

Table 2 lists the Globus services provided by the testbed. For each service the table reports a brief description of its functionality and the software module to which it belongs to. Table 3 reports the location of services inside the testbed. For each of the services the table shows the main HW/SW characteristics and the location of the host machine. As shown by these tables, the testbed involves both Intel/Linux and Sun/Solaris stations. Though similar, at first glance, the operating system kernels do not always have the same behaviour when running Globus services. Releases 7.0 and 7.1 of Linux/Red Hat as well as releases 2.7 and 2.8 of Unix/Solaris coexist within the testbed. It is worth noticing that at the time the testbed was deployed, Globus installation was not documented with respect to any 64 bits architecture. In particular no documentation was available for the release 2.8 of Unix/Solaris. Finally, it is worth noticing that the testbed is not exclusively devoted to experimental activities, but it also hosts services offered to "external" users (mail, file sharing, etc.). In this way we reproduce both a production and a controlled laboratory environment. From a practical point of view, we let the testbed evolve autonomously and, at the same time, we are able to monitor its status while performing the tests.

In order to gain access to the grid, each user must enroll for a digital certificate, following the procedures required by the Globus Certification Authority, Globus CA in what follows, by:

1. generating a pair of public/private keys and storing the private key, which is parametrized by a pass phrase, in his/her home directory;
2. sending a certificate request, containing the public key, to the Globus CA. The request must be sent from a valid, i.e. not generic, mail account;
3. storing the certificate sent back by Globus CA in his/her home directory.

This is a necessary condition, but not sufficient to get access to grid resources. Actually, as local security policies must be enforced, grid administrators must grant grid users explicit access to local systems resources, that is they must be added in their grid-mapfiles. In our testbed grid-mapfile files are managed by sites administrators in a coordinated way.

Service Name	Service Acronym	Software Module	Service Description
Gatekeeper	-	GRAM/GSI	Access control and resource allocation
Grid Indexing Information System	GIIS	MDS	Centralized database of resources information
Globus Resource Information Service	GRIS	GRAM	Publishing of resource information

TABLE 2: Globus Services Provided by the Testbed.

Provided Services	Host Main Characteristics	Host OS	Host Location
Gatekeeper, GISS, GRIS	Pentium III	Linux RedHat 7.1	Netlab Site
Gatekeeper, GRIS	Sun Ultra 5	Unix Solaris 2.8	Netlab Site
Gatekeeper, NFS, NTP	Sun Ultra 10	Unix Solaris 2.7	Netlab Site
Gatekeeper, GISS, GRIS	Pentium III	Linux RedHat 7.0	Headquarter Site
Gatekeeper, GRIS	Sun Ultra 1	Unix Solaris 2.7	Headquarter Site

TABLE 3: Services Location in the Testbed.

Problem Description	Ascribable To	Corrective Actions
Gatekeeper does not start up	OS	Fixing of configuration files
GASS does not behave properly	GT/OS	Fixing of source code
Mutual authentication fails because of NAT	GT/TS	Fixing of configuration files
Service traffic blocked by firewall	TS	Network traffic enabled among grid resources
Mutual authentication failure due to different internal DNS schema	Sun Ultra 1	DNS address traslation by hosts files

TABLE 4: List of Problems Encountered in the Testbed Deployment.

While deploying the testbed, several problems have been encountered and solved. Table 4 shows the most relevant of them. For each problem the table reports: the nature of the problem, if it is ascribable to the operating system (OS), Globus Toolkit (GT) or the testbed structure (TS), and which has been the corrective adopted action. Fixing these problems resulted in the release of a specific package containing patches and upgrades of Globus for the releases 7.0 and 7.1 of Linux/Redhat and 2.7 and 2.8 of Unix/Solaris, together with a set of additional components not required by the toolkit but useful to the extent of the present work and recommended for every production environment. In what follows we give a brief description of these components and their relevance in the deployment of the testbed.

As GSI makes use of temporary credentials, time on grid resources should be synchronized in order to minimize the possibility of mutual authentication failures due to systems clock inconsistencies. With this aim, we automate the synchronization procedures by the use of the Network Time Protocol (NTP) [18]. Thus it is possible for a host in the testbed to act as a time server, once connected to a public time synchronization service.

Globus installation and configuration procedures are long and tedious, and the full package takes up a considerable amount of disk space. For these reasons, the client components of Globus have been installed on a distributed file system: once opportunely configured the user working environment, only the shared directories have to be mounted, in order to start a grid session.

Finally, an effective monitoring system is a crucial aspect for both services and data availability. Since HBM has limited functionality, and does not offer support for critical events such as Grid services availability, disk occupation, CPU usage, number of running processes and number of connected users, its use has been deprecated by the Grid community. In our testbed HBM has been replaced by Netsaint (release 0.0.7) [19], a flexible open source application able to show monitoring information via a web server interface.

4. TESTING ACTIVITY

To better understand the significance of our plan of tests, it is worth reminding the modalities to log on a grid. Each time a user signs on the grid, a temporary certificate, proxy in what follows, is created. This operation requires the pass phrase associated to the user private key. The proxy certificate is then presented by every authentication procedure, and contains information about the certificate of the issuing user. This is particularly relevant whenever a proxy expires during the execution of a job. In this case, in fact, the only way to access the results of the job execution, is by creating a new proxy for the same user. This is not a so unusual event, since, while 8-12 hours is a reasonable time-to-live for a proxy, a data or CPU intensive computation could last for days or weeks.

Our tests have been finalized to investigate the implementation correctness and effectiveness of the following functionalities provided by Globus:

- User Authentication;
- Proxy Management;
- Access Control and Non Repudiation;
- Communication Confidentiality;
- Communication Integrity;
- Data and Services Availability.

To reach our goal, we define a set of significant procedures that solicit the target functionality. Each procedure consists in the execution of a sequence of Globus commands. Such commands are listed in Table 5, together with a brief description of their behavior, while Table 6 associates to each investigated functionality the corresponding test procedure. The evaluation method compares the behavior expected for each tested functionality with the experimental results. The expected behavior for a given GSI functionality is a behavior compliant with the standards, recommendations and proposals for the release 1.1.3 of Globus.

While the current version of GSI does not support *Communication Confidentiality* features, the implementation correctness of each of the remaining functionality is tested by one or more series of tests, each series consisting in 10 repetitions of the same procedure.

Command	Description
grid-proxy-init	Generates A New Proxy
grid-proxy-info	Returns Info About Proxy Status
grid-proxy-destroy	Terminates The Proxy
globusrun	Executes Remote Commands
globus-job-run	Executes Resource Specification Language Job Descriptions
globus-job-submit	Executes Remote Batch Commands
globus-job-get-output	Retrieves The Output Generated By A globus-job-submit Execution
globus-job-cancel	Aborts The Execution Of A globus-job-submission Command

TABLE 5: Globus Services Globus Commands in the Testing Sequences

Functionality	Sequence
User Authentication	grid-proxy-init, grid-proxy-info
Proxy Management	grid-proxy-init, globusrun, globus-job-run, globus-job-submit, globus-job-get-output
Access Control	grid-proxy-init, globusrun, globus-job-run, globus-job-submit, globus-job-get-output, grid-proxy-destroy
Non Repudiation	grid-proxy-init, globusrun, globus-job-run, globus-job-submit
Communication Integrity	grid-proxy-init, globusrun, globus-job-run, globus-job-submit
Data Availability	grid-proxy-init, globusrun, globus-job-run, globus-job-submit, globus-job-get-output, globus-job-cancel

TABLE 6: Testing Procedures.

The tests have been accomplished in both monosite (NetLab) and multisite (Netlab and Headquarter) scenarios. As far as *User Authentication* is concerned, we planned two series of tests. The first one is aimed to verify if authorized users are recognized by the security system. The second one verifies if certificate and or private key tampering attempts are correctly detected. The eventuality of a not correct pass phrase, a corrupted private key or certificate, and not related certificates and private keys, have been taken into account. The two accomplished series of tests resulted in satisfactory outcomes. Nevertheless, the adoption of a additional Certification Authorities is strongly recommended, as Globus certificates are suitable for a low level protection. Further details of these aspects are given in section 5.

With respect to *Proxy Management*, we planned two series of test. In particular, we successfully verify that expired proxies are no longer valid credentials to access grid services. We consider both the case in which the proxy is already expired at the time of job submission, and the case in which it is valid at that time, but expires during the remote job execution. In this situation we verified that the results generated by the job can be collected by the creation of a new proxy.

As far as *Access Control and Non Repudiation* are concerned, we planned two series of tests: the first one is aimed to verify the correct use of grid-mapfiles, i.e., only users mapped to local accounts in the grid-mapfiles can have access to grid services; the second one verifies that each user is granted the same privileges of the local system account it is mapped to. In both cases we always succeeded in identifying actions "responsibles". In other words, it has always been possible to distinguish grid users from the local users they were mapped to. The two accomplished series of tests resulted in satisfactory outcomes. Nevertheless, we detect some limits in the mechanism of grid users management. Actually, it is possible for a grid administrator to grant super user privileges to grid users inside local systems, in particular also in those administrated by someone else.

With respect to *Communication Integrity*, one series of tests has been planned and successfully accomplished. Finally, as far *Data and Services Availability* is concerned, we planned one series of tests aimed to verify the proper utilization of secure communication protocols in order for users and applications to access data and resources. In addition, we investigated the main characteristics of HBM, which, as already stated in section 3, revealed to be a not adequate monitoring system, thus making it hard to investigate on *services availability*. All functionality have been verified by an accurate analysis of the log files and of the system status, with the only exception of *Communication Integrity*, which required the examination of the generated packets flows.

Table 7 summarizes the results collected by our testing activity: the implementation of fundamental security services, as listed by section 2, is classified as not adequate if the implied risks must be reduced by appropriate countermeasures to make it suitable for a production environment; as adequate if it already provides an acceptable level of security. With respect to confidentiality, it is not implemented in the current release of GSI, and is consequently classified as not adequate. Nevertheless, the implementation of a confidentiality feature, will be available in a reasonable short time, since already planned by the Globus roadmap. These results point out that GSI can be transferred to a production environment if supported by and integrated with a series of services: an automatic tool for grid user management, a tool implementing an advanced local access control and a monitoring system for grid resources. Furthermore the integration with a Certification Authorities is strongly recommended.

Service	Evaluation	Countermeasure
Authentication	Not adequate	Integration with CA
Access Control	Not adequate	Integration with User Management and Policy Management tools
Confidentiality	Not implemented	Supported in subsequent releases
Integrity	Adequate	-
Availability	Not adequate	Integration with an Availability Monitor
Non Repudiation	Adequate	-

TABLE 7: Evaluation of GSI Services and Countermeasures to Inadequacies

5. REDUCING THE RISKS OF GSI BASED PRODUCTION ENVIRONMENTS

The experience achieved shows that in order to reduce the risks peculiar of a production environment, some additional features must be provided to Globus. In detail, we propose the integration with Certification Authorities other than Globus CA, an advanced mechanism for grid user management, an advanced tool for the management of local resources access policies, and a monitoring system for grid resources and services. The Globus certificates are suitable for a low-level protection, and they are used for demonstration purposes. This is not surprising since the certificate policy adopted by Globus [20] has the precise aim to define general rules acceptable by a large number of international organizations. Nevertheless, despite every country has their own legislation about digital certificates, there are some general rules that must be followed, which Globus CA lacks to address: a certificate policy (CP) and a certificate practice statement (CPS) must document functional, technical and legal issues related to the certification infrastructure; the Authority must publicly provide a certificate revocation list (CRL), containing certificates no more to be considered valid; the user registration service must be provided with more specific credentials than an email address in order to have a request accepted.

Our proposal is to add trusted Certification Authorities to a list contained in GSI configuration files. Recently, a specific tool has been developed to automate this task, currently in beta testing on release 2.0 of the toolkit; still, it is simpler to add little information to configuration files by hand. Anyway, this is only a partial solution, that does not address the issues related to CPs, CPSs and CRLs.

Grids are designed to manage a large and dynamic number of users. Consequently, there is a need to manage efficiently updates on access control files. At the state of the art, grid administrators must manually add, delete or change the links between grid certificates and local users on grid-mapfiles, on each resource joining the grid. As users are identified by means of their certificates, it is possible to store them on a centralized repository acting as a server, together with other information regarding authentication and authorization processes. Information can be stored in a LDAP directory using the same name space and schema of MDS, thus making it possible to query the server by the same commands provided by Globus for information indexing. An administrative role should periodically update the directory, while an administration tool should be able to download certificates, check certificates validity by means of CRLs, build and replicate or update configuration files on grid resources, according to a given policy (all, a specific group, domain, etc.). Automated procedures can be conceived making it possible for the Certification Authority to publish certificates on the LDAP server, and to dynamically manage configuration files, taking advantages of data stored on the server. Note that, even if the information exchanged is not particularly sensitive, a security policy should provide a controlled access to the directory service in order to avoid attacks aiming at the registration of forged certificates.

A particular care should be taken in assigning administrative roles. A grid administrator may not be granted superuser privileges on a local system: this is very likely in complex environments. Access control on grid resources, as defined in Globus, presents a serious flaw: the one who has the privilege to change the grid-mapfile, has also the power to map grid users as superusers. An effective solution could control the invocations of *grid-mapfile-addentry* and *grid-mapfile-delete-entry* commands, used respectively to add and remove users from the grid-mapfiles. A script should extract relevant user information, such as user and group identifier, and take appropriate actions with respect to the policy to enforce.

Dealing with a security infrastructure, we cannot leave availability of resources and services out of consideration; yet no valid availability model can be designed without effective monitoring mechanisms. Globus HBM only tracks limited variations of services availability in the grid, the so called "heart beats"; in addition, the same

information is maintained by the MDS, so HBM is felt as redundant and not addressing monitoring properly. The above considerations clearly imply that an ideal candidate to solve the issues concerning data and services monitoring in a grid environment should:

- be an *open source* application;
- support LDAP;
- supply a software development kit enabling software agents design;
- not be much intrusive, from clients perspective, and easily integrable in a grid development environment.

In our opinion is *Nagios* [22], in the former releases known as *Netsaint*, meets all these requirements. Nagios is an open source software whose SDK allows building functionality by means of *plugin* modules. Furthermore it matches the European choices with respect to grid monitoring issues, even if the functional areas of resources and services discovery, automatic monitoring configuration, and hardware and software inventories must be improved. Finally, it is worth noticing that several alternative approaches to Nagios have been adopted and tested by relevant projects. Among them Ganglia [23], received a considerable consensus by the U.S. grid community.

6. CONCLUSIONS AND FURTHER WORKS

Computational Grids enable the creation of a virtual computing environment for sharing and aggregation of distributed resources aimed to solve large-scale problems in science and engineering. In this paper we investigate the functional correctness and effectiveness of the GSI features with respect to the main security services peculiar of a production environment. Our study and experimentations point out that, while some issues are well addressed, others are problematic and lead us to the definition of proposals to respond to a series of inadequacies.

We conclude that the security model offered by Globus is valid, but, in order to be transferred to a production environment, it must be supported with and integrated by a series of countermeasures to increase the provided security. Some of these countermeasures have been identified and guidelines to their implementation provided.

Further investigations are on-going. We are mainly focused on the evaluation of more recent releases of the Globus Toolkit embedded in the middleware developed by the European DataGrid project, that is very close to be implemented in a real production environment.

ACKNOWLEDGMENTS

We would like to thank Dr. Flavia Ferrantelli for commenting on improving the paper and Dr. Angelo Olivieri for document drawing. We acknowledge Netlab for providing access to resources used for the present work and, among collaborators, especially Dr. Silvio Scipioni and Dr. Stefano Colica, for their work in the deployment of the testbed.

REFERENCES

- [1] I. Foster, C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1999.
- [2] I. Foster, C. Kesselmann, S. Tuecke, "The Anatomy of Grid", International J. Supercomputer Applications, 15(3), 2001.
- [3] <http://www.globus.org/research/papers.html>
- [4] <http://legion.virginia.edu/>
- [5] <http://icl.cs.utk.edu/netsolve/>
- [6] <http://www.unicore.de/>
- [7] <http://www.globus.org/>
- [8] The UK ITSEC Scheme, "UK IT Security Evaluation and Certification Scheme", <http://www.cesg.gov.uk/assurance/iacs/itsec/documents/formal-docs/media/UKSP01.PDF>.
- [9] W. S. Stallings, "Network security", Second Edition, Prentice-Hall, 2000
- [10] J. Linn, "Generic Security Service Application Program Interface Version 2, Update 1", Gennaio 2000.

- S. Meder, S. Tuecke, D. Engert, "GSS-API extensions", Internet Draft
- [11] I. Foster, N. Karonis, C. Kesselman, G. Koenig, S. Tuecke, "A Secure Communications Infrastructure for High-Performance Distributed Computing", 6th IEEE Symp. on High-Performance Distributed Computing (pag. 125-136), 1997.
- [12] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids", Proc. 5th ACM Conference on Computer and Communications Security, pp. 83-92, 1998.
- [13] K. Jackson, S. Tuecke, D. Engert, "TLS Delegation Protocol", Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-tls-delegation-01.txt>, Luglio 2001.
- [14] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)", Settembre 1993.
- [15] R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", January 1999.
- [16] <http://www.openssl.org/>
- [17] <http://www-fp.globus.org/hbm/>, Luglio 2001.
- [18] D. Mills, "Network Time Protocol (Version 3) Specification, Implementation", March 1992.
- [19] <http://www.netsaint.org/>
- [20] The Globus Project, "Globus System Security Policy and Requirement: v 0.2", www.globus.org/security/policy.html.
- [21] E. Rescorla, "Http over TLS", May 2000
- [22] <http://www.nagios.org>
- [23] <http://ganglia.sourceforge.net>

© M. Draoli, C. Gaibisso, D.Giannelli 2002