Security Mechanisms in Grid Environments

Adrian Bagala, Roland Bott

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovak Republic
{bagala, bott}@fiit.stuba.sk

Abstract. This document describes security technologies and mechanisms used in grid environments. The Grid Security Infrastructure – GSI, which is implemented in the Globus Toolkit as well, is described in detail. The main focus is on methods of identification, authentication and authorization, based on X.509 certificates and SSL/TLS protocols. Finally a solution of community-based access control over the grid resources is presented, which is interpreted on the implementation of the Globus Toolkit.

1 Introduction

Computational grid is a technology allowing IT resources of various organizations, institutions or individuals to be commonly used, creating one unified structure. It is used mainly for complicated and time consuming computation tasks. These computations run within the environment mentioned above are known as grid computing.

The infrastructure appears for the user as a single powerful computational system. So the grid is an abstraction allowing transparent and easy access to distributed computational resources. It consists of a number of interconnected resources like computational systems, data storages and the connections between these systems. The entities of this structure could be geographically spread and the interconnections are provided by network infrastructures.

The required features of the grid are security, reliability, effectiveness, low price and high throughput for computational applications. This article is focused mainly on the current most used security technologies implemented in the most grid environments.

* Supervisor: doc. Ing. Ladislav Hudec, PhD., Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava
such as the well known Globus Toolkit. The security is mainly focused on the identification, authentication and authorization of the grid users. These security technologies are similarly used in community solutions of individual access as well.

2 Grid Security Infrastructure (GSI)

The Grid Security Infrastructure was implemented as a component of the Globus Toolkit, providing security mechanisms. GSI complies with the Generic Security Service Application Programming Interface (GSS-API, RFC 2078/2743) standard. The implementation uses X.509 certificates and OpenSSL libraries [2]. The basic requirements on GSI are [2] [1]:

- Every entity or subject accessing grid resources (user, device, resource, process or application) must have its unique identity. The identity of the subject could be represented using certificates. These certificates are issued by a trusted Certification Authority (CA). Globus Toolkit uses X.509v3 certificates.
- The identity of a subject needs to be ensured or authenticated. This is provided by the TLS authentication protocol (descendant of the SSLv3 protocol). The identification information is guaranteed with the trust in CA and its signature policy.
- The subject should be allowed to run processes on remote resources as well, which are also part of the grid and to which the subject has been granted access. The entity which acts behalf of a user on a remote resource is called proxy. Globus generates a proxy with new own certificate valid only for limited time, signed by the user’s certificate. The remote device could verify the proxy certificate by the user’s signature, which is in turn verified by the Certification Authority’s signature trusted by the remote device. This way the remote device also authenticated the user standing behind the proxy certificate.
- In a grid, there could be a number of processes created for the computation, which could demand access on remote resources. For these processes belonging to one user, series of proxy certificates are created, which make possible for them to identify and authenticate themselves.
- Every resource is able to determine (authorize) if it can accept a particular incoming request, based on Access Control Lists (ACL).
- As the last step to the authorized usage of a resource after authentication is mapping the global identifier or name of a user to local. The global name is based on the subject of the proxy certificate in the format of X.500 Distinguished Name (DN). The grid resources have public gridmaps, which contains the mappings between DNs and local usernames. This method assures that the requester gets all relevant user rights.
2.1 X.509 Certificates

GSI uses for authentication purposes certificates of public keys X.509 and SSL. These certificates assign to grid subjects or entities unambiguous names and are signed by Certification Authority [5].

The format of the certificate used in the Internet is described in RFC-3280 [6]. The certificate consists of the following basic parts:

- number and version of the certificate
- name and ID of the subject and the issuer
- public key of the subject
- the validity period of the certificate
- extensions of the certificate
- the identifier of the signing algorithm
- digital signature of the CA

2.2 X.509 Proxy certificates

The proxy certificate is based on the X.509 certificate. It was set up to help perform a job process on a remote resource, to which a user has been granted access within the grid. For numerous processes a chain of short-term proxy certificates is created, also known as user proxy. Processes, belonging to one user but running on various remote resources, using these proxy certificates authenticates themselves without the need of an interactive user authentication. The certificate is signed by the user itself or by one of the user proxies. The process of remote proxy creation is called delegation, where a new private key is created along with the certificate of the corresponding public key, which is signed by the key from the creator’s X.509 certificate.

2.3 Single Sign-On

The user’s private key with long-term access rights attached is protected using various methods, which needs manual authentication. This process is needed to protect the private key. But this could be graceless when the user needs to access the key all the time, using it for authentication on remote resources [4].

The proxy certificate is addressing this problem, enabling sign-on. It makes possible for the user to authenticate manually only once at the proxy certificate creation. This proxy certificate could be used repeatedly for further authentication for a limited time period.

The creation process of the simple sign-on proxy certificate is illustrated on the following figure.
Fig. 1. Creation of a proxy certificate for single sign-on

Steps of creating a proxy certificate:

1. A new key pair which consists of a public and a private key is generated to be used in the proxy certificate. The public key is encoded in the certificate request for further processing.

2. Using the private key of the user associated with his long-term public key from his certificate the proxy certificate is signed, containing the public key from the newly generated key pair.

3. The proxy certificate and the associated private key are stored in a file. This file is protected only by the local file system.

When the proxy certificate expires, this process is repeated and the user generates a new key pair and a new proxy certificate.

2.4 Delegating through network

Proxy certificates among others could be created for delegating owner permissions for remote resources. This delegation is done using network connections. Therefore the delegation process requires the network connection to be secured against attackers [4].
Fig. 2. Delegation of a proxy certificate over a secured network connection

The figure above shows the steps of the proxy certificate created privileges delegation using network connection:

1. At the beginning host “A” on the left side is contacting the destination service “B” on the right. The initiator and the destination service are authenticating each other. The initiator uses for the authentication its existing proxy certificate and the destination service uses its certificate with its public key. After the authentication a secure connection is created, for example using the SSL/TLS protocol.

2. The initiator sends its delegation request for a particular application and the destination service generates a new key pair.

3. A certification request is signed using the new public key and sent back to the initiator using the secured connection.

4. The initiator uses the private key associated with its proxy certificate to sign the certification request. It generates a new proxy certificate including the newly generated public key from the destination service and fills in the relevant fields in the new proxy certificate.

5. The new proxy certificate is sent back to the destination service using the secure connection. The destination service saves it to a file with the generated private key. This new proxy certificate could be used on the destination service by the user’s applications.

3 Community Authorization Service (CAS)

Community Authorization Service is a new component in the Globus Toolkit version 3.2 [6]. CAS makes possible to resource providers to create identical access policy for every member of a community or group of users [3].
The CAS service enables group-based processing of access rights for members of a particular community. This needs CAS server to be created, which serves as a processor for request from community members to access provided computational resources. Using such CAS server the resource providers can define access policy for a particular community.

Every community of grid users initializes their own CAS server. The representative of the community gains GSI authorization to represent the community as a whole and deploys CAS server, which uses the community identity.

The resource providers assign access rights to a particular community of grid users, instead of each user independently. Every resource provider ensures or authorizes, that the holder of the community permission is representing that particular community and if the community policy is consistent with the resource provider’s policy.

The representatives of a community are using CAS for managing and controlling trust relationships (for example to register users and resource providers with a community using a community standard) and creating fine-grained resource access control. Community members with the appropriate right could authorize other members of the community.

When the grid user wants to access a grid resource which is accessible through a CAS server, the user creates a request to the CAS server. If the CAS database on the server indicates that the user has the required rights, then it issues a limited GSI proxy certificate for the user. This proxy certificate has an access policy attached, which grants to the user rights to fulfill the requested action.

![Fig. 3. Community based access control to resources](image-url)
The user in turn uses the certificate from the CAS server which grants community access rights to contact resources involving grid tools (for example GridFTP). The grid resource then applies its local security policy to determine the access level assigned to the community and other restrictions which are based on the security policy in the CAS certificate.

4 Conclusions

The presented security techniques are making an essential part for the future of the ever growing grid computing technology. Also while delegated proxy certificates and community authorization successfully challenges some of the grid security problems, these techniques are not fully completed yet and there is further work needed to challenge all security issues specific for grid computing.

Acknowledgement: This work was supported by Slovak Science Grant Agency grant No. VG 1/0157/03 entitled Methods and tools for development of the secure networked and distributed mobile computer systems and their management II.

References