A Data Grid Security System Based on Shared Context*

Nong Xiao, Xiaonian Wu, Wei Fu, Xiangli Qu

School of Computer Science, National University of Defense Technology,
410073 Changsha, China
xiao-n@vip.sina.com

Abstract. Data grid system supports uniform and secure access of heterogeneous distributed data resources across a range of administrative domains, each with its own local security policy. The security challenge has been a focus in a data grid environment. This paper mainly presents GridDaEn's security mechanisms. In addition to the basic authentication and authorization functionality, it provides an integrated security strategy featured by shared context-based secure channel building to leverage security processing efficiency so as to improve interaction performance occurring among multiple domains in GridDaEn. Meanwhile, by means of proxy credential single-sign-on across multiple domains can be achieved. Experiments show that this approach can guarantee system security and reliability with great performance enhancement.

1 Introduction

Data grid system integrates and manages heterogeneous distributed data resources across a range of administrative domains, each with its own local security policy. The system implies a major security challenge while providing conveniences.

GridDaEn[1](Grid Data Engine) is a data grid middleware, it is implemented by NUDT (National University of Defense Technology). The system is faced with some noted security problems in data grid circumstances such as integration, interoperability and trust problems, which greatly complicate system security mechanisms. According to our grid application backgrounds, we adopt GSI[2](Globus Security Infrastructure) as a basic framework and improve it with the introduction of shared context. GridDaEn's security mechanism is built in combination with PKI[3] (Public Key Infrastructure), with the following features included:

- Supporting mutual authentication and communication confidentiality;
- Supporting a fine-grained RBAC authorization mechanism;
- Supporting single sign-on;
- Supporting shared context-based secure channel building to improve performance.

The rest of the paper is organized as follows: Section 2 presents a brief introduction of GridDaEn system. Section 3 describes the structure and features of GridDaEn

* This paper was supported by the National 863 High Technology Plan of China under the grant No. 2002AA131010, and the National Natural Science Foundation of China under the grant No. 60203016.

security system in detail. Section 4 shows the implementation. Section 5 exhibits the performance improvement by comparing the time overheads in building secure channels with two different approaches. Section 6 introduces some related work on grid security. Finally, in Sections 7 we present our conclusions and future work.

2 GridDaEn System Overview

2.1 GridDaEn Structure Model

GridDaEn (Grid Data Engine) system is a Data Grid middleware, which can integrate various kinds of file systems and provides uniform seamless access to distributed datasets. GridDaEn consists of four major components: Client tools, Security and System manager, DRB (Data Request Broker) servers, and MDIS (Metadata Information Service), as is illustrated in figure 1.

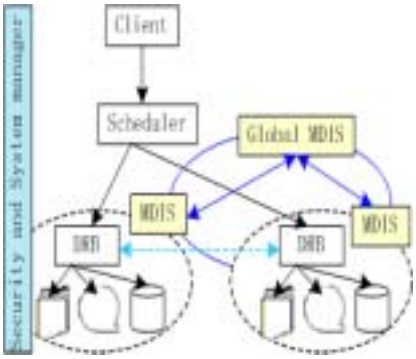

Fig. 1. The structure of GridDaEn

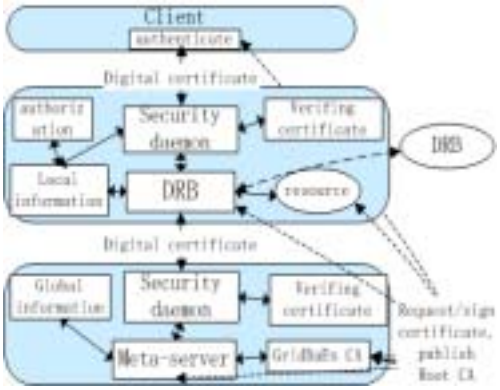

Fig. 2. GridDaEn structure with security mechanism

There are more than one administrative domains in GridDaEn. In each domain, there is a DRB server, which performs actual data operations on local storage resources in responding to requests from users and applications. MDIS, which provides metadata service for each DRB server, is organized into a distributed structure, including several local metadata servers and a global metadata server. Security information such as authorization information is partly stored in MDIS.

2.2 A Job-flow across Multi-domains in GridDaEn

In order to illustrate the mechanisms implemented in GridDaEn, first, we will analyze a typical job flow scenario across multiple domains in GridDaEn, as is demonstrated in Figure 3.

- A user contacts DRB A, and submits a job to DRB A
- DRB A contacts its MDIS M, in cooperation with which it checks the user's rights
- DRB A locates the required data. If it is located in the local domain, DRB A will contacts site S where the data resides, and then obtain data from site S. Otherwise, DRB A will inquire of MDIS M about where the data resides, then find its broker, for example, DRB B
- DRB A then contacts DRB B, and delivers the original job to DRB B
- DRB B contacts its MDIS N, checks the user's rights, and then locates the Resource site T
- DRB B contacts site T, and obtains the required data
- DRB B returns the data to the user via DRB A

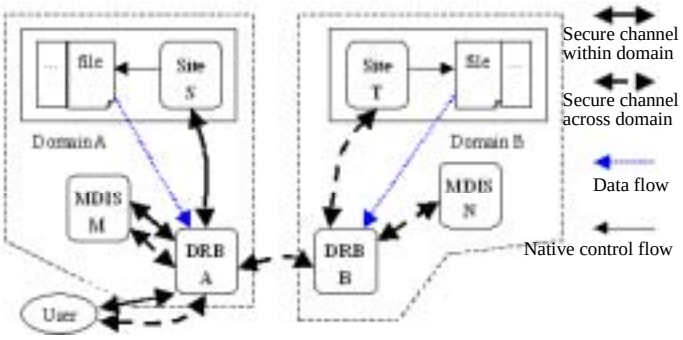

Fig. 3. Job workflow

3 GridDaEn Security Mechanisms

3.1 GridDaEn Security Structure

As is mentioned before, it is necessary to provide many security functionalities such as authentication, authorization, and communication confidentiality to guarantee security in GridDaEn. To meet the security requirements raised in GridDaEn, such functionalities as GridDaEn CA (Certificate Authority), mutual authentication, authorization, communication confidentiality etc. are mainly provided. Meanwhile, single sign-on is realized by means of proxy credentials. The security structure of GridDaEn is illustrated in Figure 2.

In Figure 2, there is a CA located at the site where GridDaEn Global MDIS resides. GridDaEn entities, such as users, DRBs, MDISes and resource sites, should request and obtain a digital certificate to identify themselves from CA.

3.2 Features of GridDaEn Security System

As can be seen from Figure 3 in Section 2.2, for security guarantee, any two participants should do mutual authentication to build a secure channel before the job can start. It is clear that, if the required data is within DRB A's domain, the system must build three secure channels (as denoted by the solid lines). Otherwise, five channels (as denoted by the dashed lines) must be built. If the job is somehow more complex, the amount of secure channels will be even larger. As a result, a large amount of time is consumed in building so many secure channels. For performance considerations, it is necessary to minimize the number of secure channels. Therefore, shared context-based secure channels are introduced to be a solution. Besides this distinct feature, other security mechanisms, such as single sign-on, RBAC authorization and so on, are also well-supported.

3.2.1 Shared Context-based Secure channels

From the above analyses, we can infer that the performance of secure channels will greatly affect the performance of the whole system. Our solution is what we called a context mechanism, similar to a connection pool, which reduces the overheads (caused by security authentication) by sharing or reusing context. Such context is built by the secure channel between any two participants in GridDaEn. When a DRB is started, it will automatically start a mutual authentication process with an MDIS to build secure channels. After that, the information associated with this process will be saved in a *context*, which has many properties and methods, as is sketched in figure 4. Notice that, for the dynamic and changeable nature of grid environment, each context should have a lifetime field to specify its validity period. Within the specified lifetime, all the data transferred between this DRB and its MDIS will be encrypted and passed by this context, and the access rights is also authorized by the identity recorded in the context. When a user wants to access some data resided in a site, a context between this user and the DRB for this site will also be established.

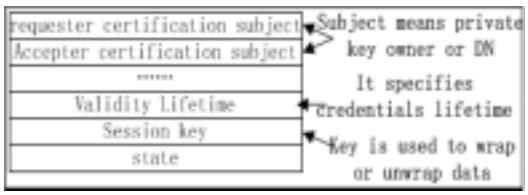

Fig. 4. The properties of a context

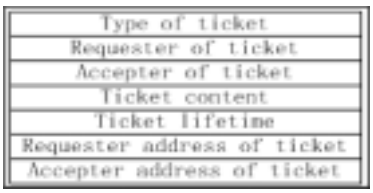

Fig. 5. A context table entry

Actually, different users cannot share the contexts between users and DRBs, that is mutual authentication must be performed again when a user wants to contact another DRB. However, those contexts between DRB and MDIS, or between DRB and resource sites, or one DRB and another DRB, can be easily shared or reused by others. Therefore, overheads will be lowered and performance will be improved..

However, how to use and manage these contexts can be a troublesome problem. Here we adopt the idea of route table. Once the system builds a context, it will be classified by its creator and saved into a table with its lifetime. The address information of the two participants is used to index this table. Figure 5 gives an example of a context table's entries. Therefore, the subsequent requests can quickly find a context and greatly accelerate its authorization process.

3.2.2 Single Sign-On and Authorization

Combined with user's proxy credentials, GridDaEn implements single sign-on. Before a user authenticates with a DRB, he will generate a proxy credential by his digital certificate. Then he authenticates himself to the DRB by this proxy. Afterwards, the proxy will execute all the activities that the user wants to do at all the sites, in complete representative of his identity.

For authorization in GridDaEn security system, we borrow some ideas from RBAC. The basic concept of RBAC[4] is that users should be assigned some roles associated with specific permissions, and users' permissions are the combination of these roles. Here we also try to introduce this flexible and effective approach into our authorization mechanism.

We have developed several tools for permission definition, role definition, and user definition. First, we define several elementary permissions for resources in our authorization mechanism. Then we define roles for each domain separately, assigning corresponding permissions to them. Note that, these roles only work in their own domains. Finally, we create grid users and assign roles to users according to some policy. If a user is assigned roles of a domain, he can perform the corresponding privileges in this domain. However, without roles of that domain, he cannot do anything. Thus, it can obtain fine-grained authorization. In order to simply authorization operations, we also introduce the notion of group, to which roles can be assigned. A user belonging to a group will inherit all the roles assigned to the group automatically. All authorization information is saved in MDIS. If authenticated, a user will be checked whether or not he possesses the specific privileges to complete his job. After that, the job will be run in identity of a local user by means of local user-mapping.

4 Implementation

GridDaEn security system is implemented in Java, therefore it can be installed on various platforms such as Linux, Windows, without any modification. We employ some functionalities of Java CoG Kit[5], such as authentication, proxy credentials generation, on which some improvement is made to meet our specific requirements.

We provide GridDaEn CA to suit for our purpose, which responds to certification requests from GridDaEn entities, also depicted in figure 2. Two participants authenticate each other using their own digital certificates. Another important module is to issue proxy certificate for each client, by means of which single sign-on can be achieved. Of course, direct authentication can also be made by digital certificate to

get higher security privilege. After authentication between any two participants, all the transferred data would be encrypted by their context to guarantee communication confidentiality. And all interfaces and APIs are standard, supported by GSS-APIs(Generic Security Service Application Program Interface).

Based on the above security functionalities, we build a context table to save contexts built after authenticated. Also it records lifetime and other information about contexts. Figure 6 illustrates a Sequence Diagram using case view, which describes how to build and share a context across multiple domains in GridDaEn. As is depicted, the context between DRB A and MDIS is built when client 1 contacts DRB A and submits job to DRB A, and it is used to process jobs from client 1, such as 5 and 6. Also it is used to process jobs from client 2, such as 14 and 15 within its lifetime. Other contexts are similar as above.

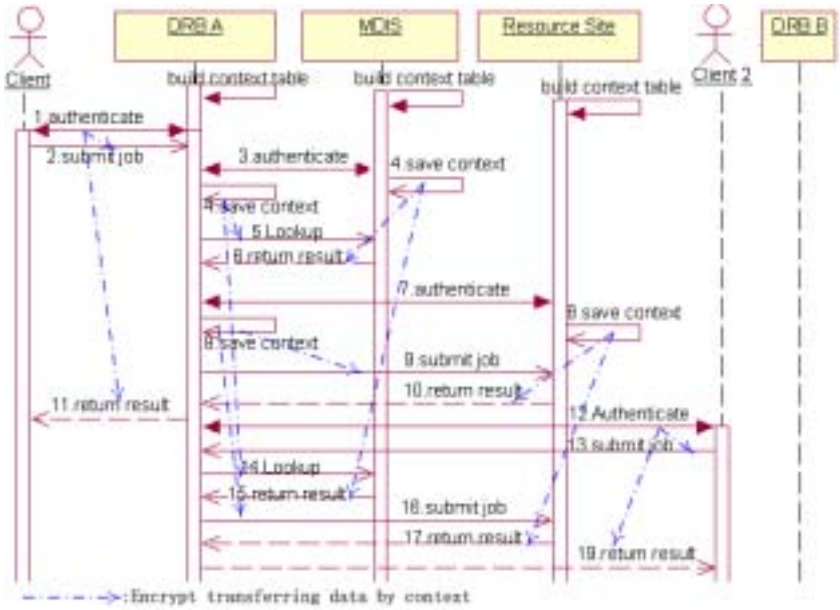

Fig. 6. A Sequence Diagram describing how to build and share context

By adopting PKI model, our security system can integrate with existing systems and technologies, and it can be deployed on all kinds of platforms or hosting environments, with distributed structure support. Meanwhile, digital certificates signed by GridDaEn CA can help to establish trust relationships among multiple domains in GridDaEn by mutual authentication.

5 Performance

For the main concern of security efficiency in our system, we have carried out some experiments to test its performance. The test environment is built with a 10Mbps hub

and four machines. The client is running on an Intel machine with 2.4GHz CPU, 512MB of main memory, 40GB Seagate IDE disk and Windows 2000 operation system. A resource site is running on a machine as above. A DRB is running on a Pentium IV machine with 2.4GHz CPU, 512MB of main memory, 40GB Seagate IDE disk and Red Hat Linux release 8.0 operation systems. An MDIS is running on a machine same as the machine running the DRB.

In one experiment, a client sends a request to DRB, reading a file from some site. Firstly, one test for mutual authentication is made, secondly, another test processing a job request for security authentication and encrypting transferred data is carried out. Time consumed in both are illustrated in Figure 7. Note that, MDIS and Resource-Site authenticate with DRB separately, and DRB authenticates with Client in addition. As can be seen from figure 7, time consumed in authentication takes a large percentage in the whole security processing (including authentication, data encryption/decryption etc.), Therefore, the performance of authentication will greatly affect the performance of the whole security system.

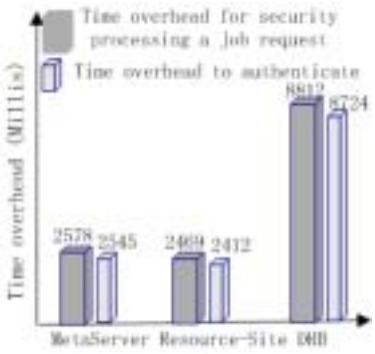

Fig. 7. Test results comparing time overheads for authentication processing only and whole security processing

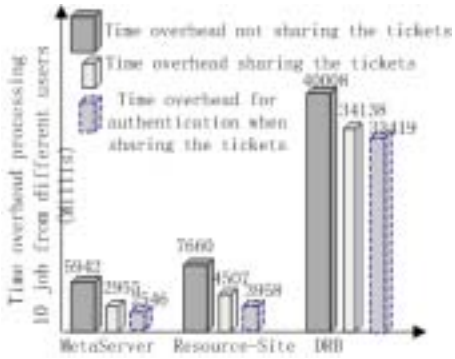

Fig. 8. Test results comparing time overheads with shared context and without shared context

In another experiment, we test security overhead in DRB, MDIS and Resource-Site when processing many jobs from clients. Firstly, a test for many job requests without shared contexts is made, and secondly, we test these jobs with shared contexts. Figure 8 illustrates the time overheads of both. Note that, with shared contexts, MDIS and Resource-Site authenticate with DRB once respectively, but DRB must authenticate with Clients each time.

From the above figures, we can see that shared contexts can improve the efficiency of the security system to a large amount.

6 Related Work

GSI is a component of the Globus Toolkit[3, 6] that has become the de-facto standard for Grid security. GSI employs PKI[4], which communicates by SSL (Secure Socket Layer), provides mutual authentication and communication confidentiality using public key cryptography and digital certification, and extends to support single sign-on. But authorization is too coarse-grained by a girdmap file.

To summarize, although existing distributed security technologies can solve relating problems faced in its domains, and provide some solutions, but which cannot adequately address the issues in our data grid environment.

7 Conclusion

In this paper, we present a new security mechanism implemented in GridDaEn system: the shared context-based security mechanism. The main contribution is that it can offer security guarantee while meeting the stringent performance requirements for GridDaEn. Currently, we are preparing for publishing the next version of this security system and totally integrating it into GridDaEn.

References

1. Nong xiao,wei fu,bin huang,xicheng lu,: Design and Implementation of data grid system GridDaEn, Computer Nation Conference of China (CNCC),2003.11
2. Luis Ferreira,Viktors Berstis,Jonathan Armstrong,et al,: Introduction to Grid Computing with Globus, http://www.ibm.com/redbooks, 2002, page 51-81
3. Internet X.509 Public Key Infrastructure：Certificate and Certificate Revocation List (CRL) Profile. http://www.ietf.org/rfc/rfc3280.txt, 2002
4. DAVID F.FERRAIOLO,RAVI SANDHU,SERBAN GAVRILA,et al,: Proposed NIST Standard for Role-Based Access Control, ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224-274.
5. Gregor von Laszewski,Ian Foster,Jarek Gawor,and Peter Lane,: A Java Commodity Grid Kit, Concurrency and Computation: Practice and Experience, vol.13, no.8-9, pp.643-662, 2001.
6. I. Foster, C. Kesselman, G. Tsudik, and and S. Tuecke,: A Security Architecture for Computational Grids, 5th ACM Conf, on Computer and Communication Security, 1997.