Procedia Computer Science

Volume 66, 2015, Pages 525–532

YSC 2015. 4th International Young Scientists Conference on Computational Science

Special aspects of the development of the security infrastructure for distributed computing systems

Julia Dubenskaya[1], Andrey Demichev[1], Alexander Kryukov[1], and Nikolay Prikhodko[2]

[1] Skobeltsyn Institute of Nuclear Physics, Lomonosov Moscow State University, Leninskie gory, 1, Moscow, 119991, Russia
jdubenskaya@gmail.com, demichev@theory.sinp.msu.ru, kryukov@theory.sinp.msu.ru

[2] Yaroslav-the-Wise Novgorod State University, Bolshaya Sank-Peterburgskaya, 41, Veliky Novgorod, 173003, Russia
nikolai.prikhodko@novsu.ru

Abstract

The paper describes some special aspects of the security infrastructure for distributed computing systems. Most heterogeneous and geographically dispersed distributed computing systems, e.g. GRID, use the Public Key Infrastructure (PKI). One of the main problem in such systems is the necessity to use time-limited proxy certificates for delegation of rights from user to an execution remote service which acts on behalf of the user. The problem is that there exists a contradiction between the limited lifetime of the proxy certificates and unpredictable time of the request processing. Our approach allows to avoid using the proxy certificates. This makes the security infrastructure of distributed computing systems simpler for development, support and use. In particular, the proxy renewal service becomes unnecessary at all.

Keywords: web service, GRID, PKI, proxy certificate, remote job submission, SaaS

1 Introduction

The development of the modern Web technology and computers sets a crucial task of building heterogeneous and geographically dispersed distributed computing systems (DCS), e.g. GRIDs, which provide users with different computing resources by means of a unified interface. The advantage of DCS is the simplification of an access to clouds, supercomputers, databases and, as consequence, growth of efficiency of scientific research and engineering developments in aero- and hydrodynamics, laser and atomic industry and in many other areas.

However, using the Web technologies for heterogeneous and geographically dispersed DCS requires more sophisticated and robust solutions for various aspects of the distributed computation in comparison with the case of local resources or more localized DCS. In particular there

Selection and peer-review under responsibility of the Scientific Programme Committee of YSC 2015

© The Authors. Published by Elsevier B.V.
doi:10.1016/j.procs.2015.11.059

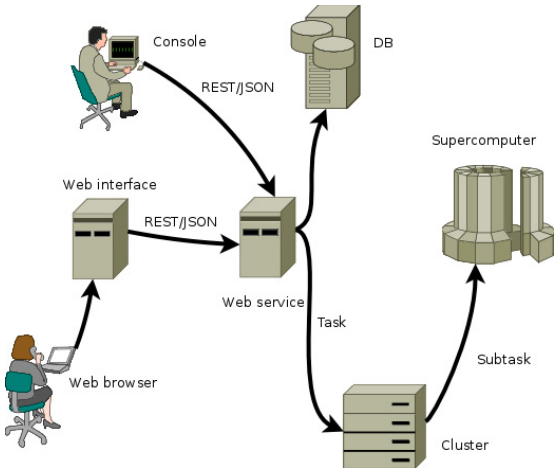

Figure 1: General scheme of the DCS

are specific aspects of a security model for such DCS. The paper describes some special aspects of the security infrastructure of such DCS and possible improvements of them. Mostly we will consider the GRID computing as a reference DCS model for implementation of the security infrastructure.

There are the two main problems which must be resolved by the DCS security infrastructure. One problem is a security of communications and another is a delegation of authority from one service to others during processing of user requests. The first problem is solved by encrypting communication channels. In the present paper we do not consider this problem and concentrate on another problem mentioned above.

For the purpose of this paper we will consider DCS as a set of interacting Web services which send requests to each other. Access points for such a system are user Web interfaces or command line interfaces installed on PCs. For example, the typical structure of DCS [1] [2] is presented in figure 1.

Providing the security of DCS implies solving the following basic problems:

1. Authentication. This means confirmation of the truth of an attribute of a single piece of data claimed true by an entity.

2. Authorization. This means the granting of access rights according to a policy. For example, each virtual organization(VO) in GRID has certain policy for resource access.

3. Delegation. This means delegation of rights from a user or a Web service to an executing Web service.

In this paper we will consider the last aspect of the DCS security by using GRID systems as an example. However the same problems are relevant and the suggested solutions are applicable for any DCS which comprises a set of communicating Web services.

One of the most successful and the biggest DCS is the European GRID infrastructure EGI [3] which is based on the gLite middleware [4, 5]. The main part of the European GRID is the Worldwide LHC Computing GRID (WLCG) [6] which is used for processing and simulation of experimental data from the Large Hadron Collider (LHC) [7]. The fantastic result of the LHC

experiments is the discovery of Higgs boson [8] which could not be reached without the WLCG [9].

The security infrastructure of the WLCG is based on the PKI [10] and X.509 certificates [11]. A user signs a request to the WLCG by a self signed secondary certificated, namely by the proxy certificate. For security reasons the proxy certificate has short life time in opposite to the user certificate which is valid for about one year. The GRID peculiarity is unpredictable time of a request execution because the system contains a number of queues where the request may turn out in a pending state. Thus proxy certificate may expire and this causes request execution failure. To prevent such a situation, the MyProxy [**?**] service is installed to renew proxy certificates if necessary.

The necessity to use the MyProxy service significantly complicates the security infrastructure. First, it is necessary to use additional service. Secondly, users have to generate proxy certificates and load them into the MyProxy. Thirdly, the renew procedure is not a simple one. All of these points creates a lot of difficulties for developers, system administrators and users.

The main idea of the proposed approach is to omit using the proxy certificates in security infrastructures at all. Roughly speaking, in our scheme each issued request is a pair of a message and individual hash related to it. This single-shot hash has unlimited lifetime. Consequently, in the new scheme, such service as MyProxy is not needed.

We would like to emphasize that the unlimited lifetime of the hash is compensated by the fact that it can be used only once and only for a specific query. Thus hash compromise can only result in the fact that the request can be processed again. It is not too much to pay for the improvements that our approach will provide.

Although the examples are given in the context of GRIDs, our suggestion is applicable to any DCS that can be considered as a set of interconnected Web services..

There are several approaches for addressing problems associated with short lifetime of proxy certificates. In the papers [12] [13] the authors focused on simplifying user's interaction with the MyProxy service by placing long-lived proxy into MyProxy service and use it to automatically generate short-lived proxy. However this approach does not solve the complexities of managing X.509-based certificates and command-line tools.

The approach based on the Kerberos tickets is used in the Kerberized Certificate Authority [14]. This system is built on the top of the Kerberos. It uses the tickets to generate a proxy certificate on the fly using the information on the credentials contained in the ticket. In our approach, we propose to completely refuse the proxy certificates.

Thus the analysis shows that the proposed approach should give essential advantage for GRID-like systems with heterogeneous resources including data storages, supercomputers, clouds, etc.

In the next section we will consider the general structure of the security infrastructure for DCS and present criticism of solutions currently used. In the section 3 the detailed conception of the solution without proxy certificates is presented. In Conclusion the obtained results are listed and possible future investigations are described in brief.

2 DCS security infrastructure

The general principles of the modern approach to building a GRID system is well described in [15] on the example of the Globus toolkit, version 4 [16]. The security infrastructure is build around PKI that uses asymmetric cryptography. One of the main problem of the security infrastructure is the problem of delegation of rights [17].

Let us consider the delegation procedure in DCS for the following taskflow (see fig. 2):

- A Client asks a $Service_1$ to perform a request.
- The $Service_1$ sends a subrequest to $Service_2$.

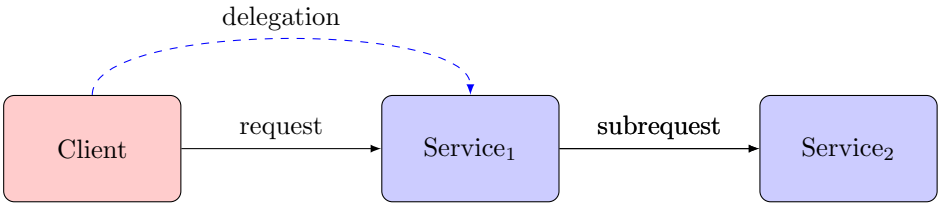

Figure 2: Delegation of credentials

It is expected that the Client somehow delegates its rights to $Service_1$ to authenticate it to the $Service_2$ since subrequest is performed on his behalf. Therefore there are a question how this delegation is is carried out.

The common solution used in GRID is to use the proxy certificate with noncritical extension such as VOMS (Virtual Organization Management Service [**?**]) extension to store information about user rights. The proxy certificate is an extended X.509 public key certificate and has the following properties:

- The proxy certificate is signed with standard X.509 of user which requires delegation of rights or another proxy certificate;
- The proxy certificate contains both public and private keys; these are not the original users keys but generated from them;
- The proxy certificate needs no password (unlike usual PKI certificates);
- The proxy certificate cannot be revoked;
- The proxy certificates are used by GRID services, to act on behalf of the proxy issuer.

Thus the proxy certificates are essentially less secure objects than standard certificates. To reduce the chance for proxy certificate to be stolen, the proxy must have very short lifetime. This leads to the problem of the renovation of the proxy. The possible solution of the problem is to use certain service that have to manage proxy certificates and renew them if necessary. One of such service is the MyProxy service.

In a common context of task execution in GRID [18] delegation and renewal process in GRID is shown in figure 3. Here 'User interface', 'WM service' and 'Execution service' correspond to Client, $Service_1$, $Service_2$ in figure 2.

3 Security infrastructure without proxy certificates

Let us consider a set of entities (services or users) which interact with each other. All these entities could be divided into two classes:

- Services and users which produce requests on behalf of themselves;
- Services which produce requests on behalf of other entities.

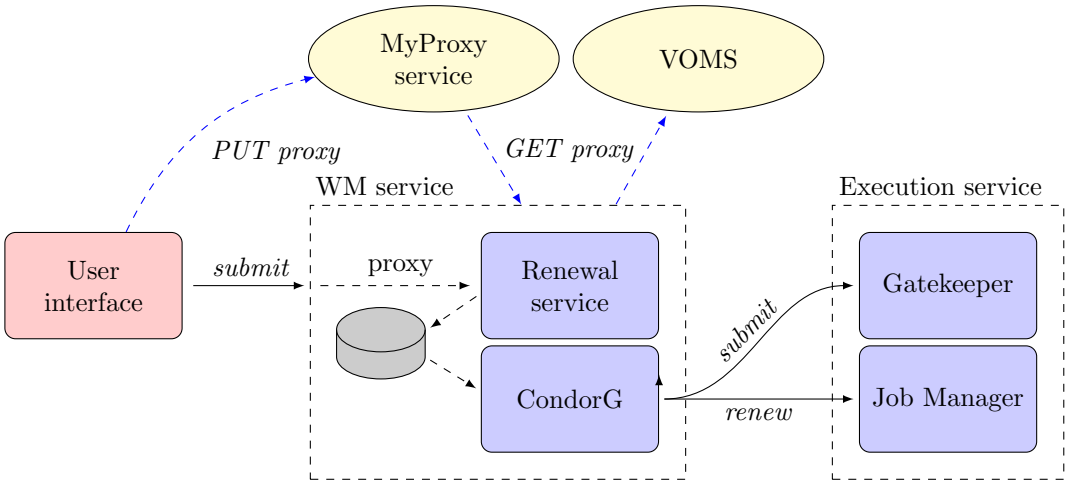

Figure 3: Renewal proxy procedure

In context of tasks execution in GRIDs, entities of first type are users or services which play a role of access points to DCS. The entities of second class are generally execution or data storage services. We also assume that all services are registered in an VS (Validation Service). This means that all requests from unknown services will be rejected.

The proposed architecture of the DCS security infrastructure is shown in figure 4.

Each request processed in DCS is accompanied by an accounting information. Accounting information is a triple of the following objects:

$$ac = \{h, Entiry_s, Entity_d\}, FIXME$$

where h, $Entity_s$, $Entity_d$ are the hash, source and destination entity of request. The triple ac means that the entity $Entity_s$ sends a request with the hash h to the entity $Entity_d$ for execution. Complete format of accounting information include some additional objects such as affiliation to a virtual organization and user's roles in it.

Let us consider the processing of a request from the point of view of the credential delegation.

1. The Client generate a request r_1 and the hash $h_1 = H(r_1)$.
2. The Client registers the triple $\{h_1, Client, Service_1\}$ in the VS.
3. The Client sends the request r_1 to the Service_1 for processing.
4. The Service_1 generates the hash from the obtained request r_1 and asks the VS to approve it. If VS approves then Service_1 continues.
5. The Service_1 generates the new subrequest r_2 that is generated from r_1 and the hash $h_2 = H(r_2)$.
6. The Service_1 registers the triple $\{h_2, Service_1, Service_2\}$ in the VS.
7. The Service_1 sends the request to the Service_2 for further processing.
8. The Service_2 generates the hash from the obtained request r_2 and asks the VS to approve it. If VS approves then Service_2 continues.

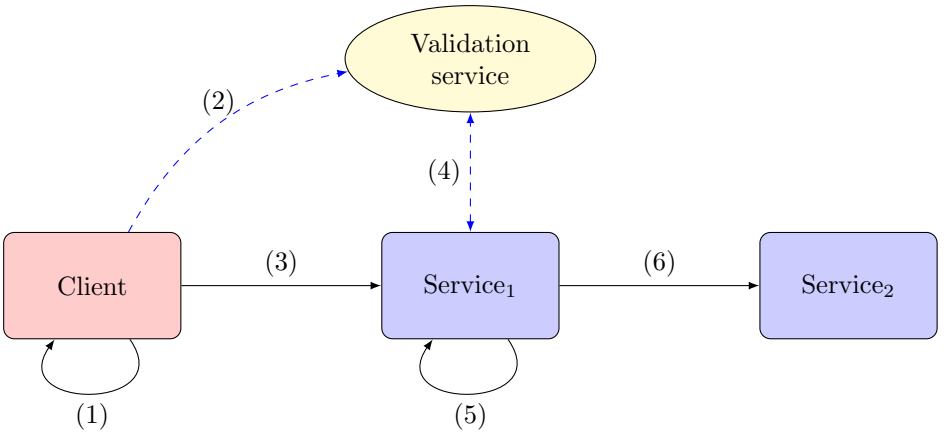

Figure 4: New architecture of security infrastructure of DCS. See details in the text.

Table 1: Comparision of standard and new approaches

Features	With proxy	New approach
Transfer both public and private keys	YES	NO
Password protection	NO	NO
Revocation	NO	YES
Acts on behalf of a user	YES	YES
Credential lifetime	very short	unlimited

In processing the request, the validation service accumulates chains of accounting information for each request in the DCS. This information can be used for different purposes. In particular, it may be used for revocation of the request at any stage of processing.

In the table 1 we compare standard approach based on proxy certificates and the proposed approach. The table shows that the new approach has several advantages over the old approach. For example, the new approach allows to revoke the request at any stage of processing. The disadvantage of this approach is the need to generate a separate hash for each request.

4 Conclusion

In this brief article we present a new approach to building the security infrastructure of DCS without the use of the proxy certificates with short lifetime. This approach allows to eliminate the use of credential management service such as MyProxy to simplify the development of DCS, its installation and support and, finally, interaction of users with the systems. The disadvantage of this approach is the need to generate a separate hash for each request.

Further development of this approach involves improving the revocation procedure.

Acknowledgments

This work is supported by the Ministry of Science and Education of Russia, agreement No.14.604.21.0146.

References

[1] D. Georgakopoulos and M. P. Papazoglou, eds., *Service-Oriented Computing*. Cambridge, Massachusetts, London, England: The MIT Press, 2009.

[2] G. Alonso, Casati, H. F., Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Springer, 2004.

[3] D. Kranzlmüller, J. M. Lucas, and P. Öster, "The European Grid Initiative (EGI)," in *Remote Instrumentation and Virtual Laboratories* (F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, eds.), ch. 6, pp. 61–66, Boston, MA: Springer US, 2010.

[4] E. Laure, C. Gr, S. Fisher, A. Frohner, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, R. Byrom, L. Cornwall, M. Craig, A. D. Meglio, A. Djaoui, F. Giacomini, J. Hahkala, F. Hemmer, S. Hicks, A. Edlund, A. Maraschini, R. Middleton, M. Sgaravatto, M. Steenbakkers, J. Walk, and A. Wilson, "Programming the grid with glite," in *Computational Methods in Science and Technology*, p. 2006, 2006.

[5] A. Garcia, "Grid: architecture and components (on the example of the glite middleware)." [online], 2006. `http://indico.cern.ch/event/431692/session/s0/attachments/935462/1325432/3_Berlich.gridcomponents.pdf`.

[6] A. Sciaba, J. Andreeva, S. Campana, F. Donno, M. Litmaath, N. Magini, J. T. Moscicki, and H. Renshall, "Computing at the Petabyte scale with the WLCG. Worldwide LHC Computing Grid," Tech. Rep. CERN-IT-Note-2010-006, CERN, Geneva, May 2010.

[7] P. Nath, B. D. Nelson, H. Davoudiasl, B. Dutta, D. Feldman, Z. Liu, T. Han, P. Langacker, R. Mohapatra, J. Valle, A. Pilaftsis, D. Zerwas, S. AbdusSalam, C. Adam-Bourdarios, J. A. Aguilar-Saavedra, B. Allanach, B. Altunkaynak, L. A. Anchordoqui, H. Baer, B. Bajc, O. Buchmueller, M. Carena, R. Cavanaugh, S. Chang, K. Choi, C. Csaki, S. Dawson, F. de Campos, A. De Roeck, M. Duhrssen, O. J. P. Eboli, J. R. Ellis, H. Flacher, H. Goldberg, W. Grimus, U. Haisch, S. Heinemeyer, M. Hirsch, M. Holmes, T. Ibrahim, G. Isidori, G. Kane, K. Kong, R. Lafaye, G. Landsberg, L. Lavoura, J. S. Lee, S. J. Lee, M. Lisanti, D. Lust, M. B. Magro, R. Mahbubani, M. Malinsky, F. Maltoni, S. Morisi, M. M. Muhlleitner, B. Mukhopadhyaya, M. Neubert, K. A. Olive, G. Perez, P. F. Perez, T. Plehn, E. Ponton, W. Porod, F. Quevedo, M. Rauch, D. Restrepo, T. G. Rizzo, J. C. Romao, F. J. Ronga, J. Santiago, J. Schechter, G. Senjanovic, J. Shao, M. Spira, S. Stieberger, Z. Sullivan, T. M. P. Tait, X. Tata, T. R. Taylor, M. Toharia, J. Wacker, C. E. M. Wagner, L. T. Wang, G. Weiglein, D. Zeppenfeld, and K. Zurek, "The Hunt for New Physics at the Large Hadron Collider," Jan. 2010.

[8] T. C. Collaboration, "A new boson with a mass of 125 gev observed with the cms experiment at the large hadron collider," *Science*, vol. 338, no. 6114, pp. 1569—1575, 2012.

[9] "Cern experiments found the long-sought higgs boson." [online], 2012. `http://www.twgrid.org/en/index.php?option=com_content&task=view&id=168`.

[10] J. A. Buchmann, E. Karatsiolis, and A. Wiesmaier, *Introduction to Public Key Infrastructures*. Springer, 2013.

[11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "RFC 5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," tech. rep., May 2008.

[12] P. Kacsuk and S. G., "Multi-grid, multi-user workflows in the p-grade portal," *Journal of Grid Computing*, no. 3.

[13] P. Kacsuk and et al., "Ws-pgrade: supporting parameter sweep applications in workflows," in *In conjunction with SC 2008*, no. 3.

[14] O. Kornievskaia, P. Honeyman, B. Doster, and K. Coffman, "Kerberized credential translation: A solution to web access control," in *USENIX Security Symposium*, 2001.

[15] I. Foster, "Globus toolkit version 4: Software for service-oriented systems," *J. Comput. Sci. and Technol.*, vol. 21, no. 4, pp. 513—520, 2006.

[16] "Globus toolkit." [online]. `http://toolkit.globus.org/toolkit/`.

[17] O. Smirnova, "Grid computing: Delegation and authorisation." [online], 2014. `http://www.hep.lu.se/courses/grid/2014/Grid-COMPUTE-l3.pdf`.

[18] D. Kouril and J. Basney., "A credential renewal service for long-running jobs," in *6th IEEE/ACM International Workshop on Grid Computing (Grid 2005), November 13-14*, pp. 2—13, 2005.