Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Systems and Software 75 (2005) 125–137
The Journal of Systems and Software
www.elsevier.com/locate/jss

Secure agent computation: X.509 Proxy Certificates in a multi-lingual agent framework

Subhashini Raghunathan *, Armin R. Mikler, Cliff Cozzolino

Department of Computer Science, University of North Texas, 225 Bryan 4, Denton, TX 76203, USA

Received 3 October 2002; received in revised form 12 January 2004; accepted 19 January 2004
Available online 19 March 2004

Abstract

Mobile agent technology presents an attractive alternative to the client–server paradigm for several network and real-time applications. However, for most applications, the lack of a viable agent security model has limited the adoption of the agent paradigm. This paper describes how the security infrastructure for computational Grids using X.509 Proxy Certificates can be extended to facilitate security for mobile agents. Proxy Certificates serve as credentials for Grid applications, and their primary purpose is the temporary delegation of authority. We are exploiting the similarities between Grid applications and mobile agent applications, and motivate the use of Proxy Certificates as credentials for mobile agents. Further, we propose extensions for Proxy Certificates to facilitate the characteristics of mobile agent applications, and present mechanisms that achieve agent-to-host authentication, restriction of agent privileges, and secure delegation of authority during spawning of new agents.

Keywords: Mobile agents; Agent security; Proxy Certificates; Authentication; Delegation; Grids; Agent infrastructure; DADS

1. Introduction

Most network-centric applications employ the client–server model. Both, client and server are static processes that communicate data over a network using message passing or remote procedure calls (RPC). Mobile agents present a paradigm shift in the way we traditionally view network or distributed applications, in that code travels over the network to where the data resides, instead of vice versa, as in the client–server paradigm. The benefits of this approach include reduced bandwidth requirements in data filtering applications, reduced network latency in highly interactive applications, and disconnected operation in mobile client computing (Harrison et al., 1995). Mobile agents are also well suited to distributed applications by virtue of their ability to clone themselves and designate a clone for each sub-task. In addition, agents are being proposed as an increasingly attractive solution for real-time applications such as routing and resource management in networks due to their reduced network overhead and fault-tolerant properties (Karnik and Tripathi, 1998).

A mobile agent, in general, is an autonomous, goal-directed software entity that acts on behalf of a user and is capable of migration from host to host in order to achieve its goal. An agent consists of a persistent code section that contains instructions, and a data section that stores the agent's state. The latter could further be divided into static data that does not change over the lifetime of the agent, and dynamic data that changes as a result of agent computation as it migrates from host to host. An agent originates on a host called its home platform, and during its lifetime, may perform various activities such as acquiring resources on a host, migrating from host to host, communicating with other agents on local or remote hosts, creating clones, or merging with other agents (Karnik and Tripathi, 1998). The itinerary of the agent may cause it to cross the boundaries of the trust domain that it originated in. This gives rise to a number of issues related to host and agent

* Corresponding author. Tel.: +1-425-705-5143; fax: +1-940-565-2799.
E-mail addresses: shubi4@hotmail.com, raghunat@cs.unt.edu (S. Raghunathan), mikler@cs.unt.edu (A.R. Mikler), cozzolin@cs.unt.edu (C. Cozzolino).

0164-1212/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2004.01.008

security (Jansen and Karygiannis, 2000; Farmer et al., 1996b).

Host security deals with protection of the host's resources and execution environment from tampering by malicious agents, while agent security deals with protection of an agent's code and state from eavesdropping or tampering by malicious hosts or network eavesdroppers. Before an agent migrates to a remote host, the agent and host must mutually authenticate each other in order to establish their identities. The migrating agent could protect the confidentiality and integrity of its code and data while in transit through the use of encryption and cryptographic checksums respectively. The host, in turn, could use techniques such as code integrity checks, code correctness proofs (Necula, 1997), and state appraisal functions (Farmer et al., 1996a) in order to verify the correctness of the agent code and state. Once the agent is accepted by the remote host and begins execution, each resource on the host requested by the agent must undergo an authorization check to ensure that the agent can be allowed access to the requested resource. Another desired security feature for agents is secure delegation. When an agent clones itself it must *delegate*, or transfer its authority to its clone. The clone in turn must have the capacity to delegate authority to any agents it creates. The transfer of authority must be done securely such that it is difficult for a malicious entity to steal credentials from, and impersonate either of the two entities involved in a delegation.

This paper presents mechanisms for authentication, authorization, and secure delegation in the agent environment. The main contribution of the work presented in this paper is the extension of security mechanisms used in Grid computing and their integration into the mobile agent paradigm. The goal of this effort is to facilitate the design of a secure mobile agent execution environment, with primary focus on the following issues:

- *Authentication:* Mechanisms by which an agent could authenticate itself to remote hosts.
- *Authorization:* Means by which an agent's rights are limited, to exactly those required to perform its task.
- *Delegation:* Mechanisms to securely delegate the originator's authority from the originator to an agent, and from one agent to another.

The security mechanisms to be extended to meet specific requirements of a mobile agent infrastructure are part of the Grid Security Infrastructure (GSI), a highly successful security architecture for Grid computing. In what follows, we will outline how this security model could be manifested inside an agent infrastructure, such as the Distributed Agent Delivery System (DADS). Section 2 introduces computational Grids, examines pertinent security issues and corresponding solutions, and motivates the application of Grid security to the mobile agent paradigm. The integration of Grid security into a mobile agent infrastructure is discussed in Section 3. This section focuses primarily on authentication of agents, restriction of agent privileges, and secure delegation of authority. Section 4 illustrates the implementation of extended GSI mechanisms into an existing agent system. Section 5 concludes the paper with a discussion of possible extensions.

2. Security for Grid applications

Computational Grids (Foster and Kesselman, 1999) are high-performance distributed computing environments that provide pervasive access to computational resources through large-scale resource sharing among multiple heterogeneous networks. Thus, Grids attempt to provide computing power "on demand" to applications, much like today's power-grids, which supply electricity on demand to consumers. The user in effect pays for using computational power but not for the cost of the computing equipment itself. The result is enormous processing power that could be used for complex scientific computation, modeling and visualization experiments, distributed data mining, and other supercomputing applications (Foster, 2001).

A Grid application differs from traditional distributed applications in several ways. It is characterized by a large and dynamic user population that shares resources across large heterogeneous networks. During the course of its lifetime, a Grid application started by a user may acquire and release resources dynamically on several machines. These resources would typically belong to different administrative and trust domains, hence it is necessary for the user to be authenticated by each of the required resources before accessing them. Further, Grid applications are usually long-lived, requiring several hours or even days to complete. Hence, it is desirable that the user authenticate herself just once at the start of the computation, and have a user process authenticate to resources on her behalf as and when required, without further intervention from her. This property is known as *single sign-on*, and it implies the need for a mechanism to delegate the user's authority to processes acting on her behalf. In addition to single sign-on, delegation is required in situations such as third-party data transfers, where a user may wish to transfer data between two remote hosts, say B and C. Instead of authenticating herself to B and reading data from B, then authenticating herself to C and writing data to C, the user could delegate her authority to both B and C, which could then mutually authenticate each other using the delegated credential and directly transfer data.

In a Grid, users and resources, by virtue of being located in different administrative domains, may employ different mechanisms for authentication such as