Access Control with Safe Role Assignment for Mobile Agents

G. Navarro, J. Borrell, J. A. Ortega-Ruiz and S. Robles
Dept. of Information and Communications Engineering
Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
{gnavarro, jborrell, jao, srobles}@ccd.uab.es

ABSTRACT

Mobile agent systems provide new perspectives for distributed e-commerce applications. Sea-of-Data (SoD) applications are those that need to process huge quantities of distributed data. They present specific restrictions, which make mobile agent systems one of the most feasible technologies to implement them. In this paper we propose a mechanism to safely assign roles to mobile agents and an access control method based in *Role-based Access Control* (RBAC). The access control method provides a simple, lightweight and distributed model for mobile agent applications. It provides a role system implemented with *Simple Public Key Infrastructure* (SPKI) certificates and uses the authorization model of SPKI for trust management.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Security

Keywords

Mobile agents, access control, security, SPKI.

1. INTRODUCTION

During the last years, mobile agent technologies have witnessed an steady, if not fast, increase in popularity. Probably, the main hurdle to a wider adoption are the security issues that mobility brings to the picture [2]. Among them, an outstanding one is resource access control. Traditional access control methods rely on the use of standard public key infrastructures (PKI) based on the authentication of global identities (via X.509 certificates). These methods allow to explicitly limit access to a given resource through attribute certificates or Access Control Lists (ACL), and rely on centralized control via a Certification Authority (CA). Despite providing effective means of protection, these techniques suffer from serious drawbacks; in particular, they give raise to closed and hardly scalable systems. Practical mobile agent systems demand lightweight, flexible and scalable solutions for access control, in order to cope with the highly heterogeneous nature of their clients. In the same vein, solutions depending on centralized entities (such as CAs) should be avoided.

Privilege Management Infrastructures (PMI) provide an alternative to PKI-based resource access control. PMI can be based on trust management and allows to assign authorizations (permissions or credentials) to concrete entities, as well as trust delegation among entities. A well-known implementation of these infrastructures is the *Simple Public Key Infrastructure* (SPKI)[3], and several security frameworks are based upon it [1]. Recent developments in this area, in an attempt to further ease access control management, have brought into the picture Role-based Access Control (RBAC) methods [4]. In these schemes, privileges of principals requesting access to a resource are determined by their membership to predefined roles.

This poster presents an application of RBAC to a concrete kind of mobile agent applications—namely, *Sea of Data* (SoD) applications. We combine RBAC and SPKI to provide a flexible, lightweight methodology for resource control in such scenarios. In our approach, mobile agents do not carry any explicit information regarding resources access, avoiding the privacy concerns associated with sensitive data embedding in mobile code. In addition, our framework allows dynamical binding of authorizations to agents, providing thus great flexibility when it comes to define resource access control policies based on it.

2. ACCESS CONTROL FOR MOBILE AGENTS

One of the first problems we found when planning the authorization model, is if the mobile agents should have SPKI keys and be considered as principals. A mobile agent cannot trivially store a private key, so it cannot perform cryptographic operations such as digital signatures. There are some propositions to store sensitive information (private keys) in mobile agents. But the problem arises when the mobile agent uses the private key to compute a cryptographic operation. The agency where the agent is in execution will

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

be able to see the private key. As a result we consider that a mobile agent should not have a private key.

Our solution is to establish the role membership of a mobile agent directly. In a way that the agent does not need to carry any kind of authorization information, making the agent more simple and lightweight.

The access control method is made up of independent modules, which interact to perform all the related tasks. These modules are implemented as static agents, they have a SPKI key and are considered as SPKI principals.

Authorization Manager (AM) manages the delegation of authorizations, issuing SPKI authorization certificates. It follows a *local authorization policy*, and its main responsibility is to delegate authorizations to the specific roles following its local policy. It may also provide the ability to delegate authorizations to other AM in order to distribute the authorization management. Since the authorization policy is local to the AM agent, it does not need to follow any specification and its format is implementation dependent.

Role Manager (RM) manages the roles (mainly the role membership) by issuing name certificates and following a *local role policy*. It can also assign a role to another role defined by itself or by another RM. Thus allowing the definition of role hierarchies or the delegation of role membership management. Each RM has a local role policy which determines what roles does it manage. It also includes rules to determine if a given principal requesting a role membership has to be granted or not. This is done by using a *membership-request*, which is equivalent to a *authorization-request*, and specifies the name certificate requested. If we choose to describe the role policy as a SPKI ACL, it is analogous to an authorization policy.

Resource Controller (RC) The Resource Controller (RC) main task is to control the access to a resource (data). It holds the master SPKI key to access the resource, delegates authorizations to AMs, and verifies that an agent requesting access to the resource has a proper authorization. It delegates authorizations to one or more AM following a local authorization policy. Note that this policy is quite simple because the main authorization management is performed by the AM.

Certificate Repository Manager (CRM) The Certificate Repository Manager (CRM) implements and manages a certificate repository, and provides services such as certificate chain discovery. For example, one agency may have one CRM to collect all the certificates issued by agents inside the agency. The CRM provides the repository and all the services needed to query, store or retrieve the certificates in the repository. It also provides a certificate chain discovery service. A principal can make a query to the CRM to find a specific certificate chain. This way we solve the problems derived from certificate distribution and leave the task to perform chain discoveries to the CRM and not to the other principals. It decreases communication traffic, certificates do not need to travel from one principal to another, and reduces the task that generic principals need to perform.

3. ESTABLISHING ROLE MEMBERSHIP

Since mobile agents cannot have private keys, we can not delegate authorizations to the mobile agent or make it member of a role. Our approach is to set as member of the role, a hash of the agent's code. The subject of a SPKI certificate and any SPKI principal in general can be a public key, a hash of a public key, or a hash of an object in general[3]. So a hash may be seen as a principal, subject of a certificate.

User-managed role The RM makes member of a given role, another role defined by a user. The user can manage its own role without intervention of the RM.

RM-managed role The RM makes member of a given role, the user. Then the users sends a request to the *RM* to set the agent code's hash as member of the role.

Note that the authorizations associated to an agent may be determined by its role membership. This way we can say that the agent will have *dynamically assigned authorizations* during its lifetime. If the authorizations associated with a role change, the authorizations related to the agent also change.

4. CONCLUSIONS

We have proposed an access control system for SoD applications, based on a mobile agent platform. It provides a simple, flexible and scalable way of controlling the access to resources. It takes the advantages of RBAC and trust management ideas. The proposed model is an extension of the MARISM-A project, a secure mobile agent platform for SoD applications.

5. ACKNOWLEDGMENTS

This work has been partially funded by the Spanish Government Commission CICYT, through its grant TIC2003-02041.

6. REFERENCES

[1] T. Aura. Distributed access-rights management with delegation certificates. In *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, LNCS 1603. 1999.

[2] D. Chess. Security issues of mobile agents. In *Mobile Agents*, volume 1477 of *LNCS*. Springer-Verlag, 1998.

[3] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. RFC 2693: SPKI certificate theory. The Internet Society, September 1999.

[4] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. In *ACM Transactions on Information and System Security*, volume 4, 2001.