Inter-domain authorization and delegation for business-to-business e-commerce.

Pietro Michiardi and Refik Molva
{First Name.Last Name}@eurecom.fr
Institut Eurécom, 2229 Route des Crêtes BP 193
06904 Sophia-Antipolis France

Abstract. Security exposures are viewed as a major impediment to the growth of electronic commerce over Internet. The main requirement of inter-enterprise communications is the verification of the role granted by a company to each individual instead of the authentication of individuals based on their universal names as provided by X509 digital ID's.
We depict in this paper an original mechanism for role-based authorization in inter-enterprise business communications. This mechanism is based on a secure extension of X509 ID certificates using SPKI authorization certificates. The mechanism was transparently integrated into existing application and network security packages. This platform was developed as part of an R&D project supported by the TEN TELECOM program of the European Commission.

1. Introduction

This paper focuses on *inter-domain access control* for business transactions carried out over Internet. Different access control mechanisms have been studied in order to find a model suitable to solve the inter-domain communications problem.

The goal of access control (AC) is to assure that interactions between an active entity called *subject* and a passive entity called *object* are authorized. Most AC systems are based on the *reference monitor* concept whereby all interactions between subjects and objects are controlled by a central entity that verifies the compliance of each communication with respect to the security policy. A security policy can be represented by an access control matrix where each cell (i, j) represents the rights of the subject associated with line i concerning the object associated with column j. The reference monitor concept can further be implemented in two different ways: *access control lists* (ACL) and *capabilities*. Another element of AC is the *security domain* that consist of the set of subjects and objects that are managed by a common security policy defined by a single *authority.*

The generic AC model based on a the reference monitor concept poses particularly challenging problems in the case of inter-domain communications, that is, when transactions take place between two different domains that are under the control of different authorities.

A suitable solution to the inter-domain AC problem is offered by the so-called Role-Based Access Control (RBAC) model [5, 6]. As opposed to the generic AC model based on a simple subject-object relation, RBAC defines the AC policy in two different relations: the subject-role relation assigning to each subject a role associated with the function of the subject in the organization and the role-object relation defining the rights granted to each role in terms of resource utilization. Thanks to its representation of the AC policy in two separate relations, the RBAC model lends itself naturally to the solution of inter-domain

AC problems. In a typical inter-domain transaction involving a subject and an object from different domains, the subject-role relation can be defined by the subject's domain authority whereas the role-object relation can be defined by the object's domain authority. Based on the separation of these two relations, the instances of each relation can be independently managed in each domain by the corresponding domain authority. Thus unlike the generic AC model, RBAC allows for the separate management of each domain.

2. Solution

The inter-domain AC solution presented in this paper was designed for web-based client-server applications. The main feature of the design is seamless integration with the existing application infrastructure. Access control modules are transparently integrated with the existing client-server platform using Java applets and servlets interfacing with the www client and server programs. Figure 1 depicts a typical inter-domain scenario involving subjects (s_i) from domain D_s and objects or resources (r_j) from domain D_r. The resources are managed by a web server in domain D_r.

2.1. Access control components

The web-server deployed in D_r is integrated with the Access Control Module (ACM) and the Domain Delegation Module (DDM). The ACM module is responsible for taking the access control decision to grant or deny access to a resource for a particular user. The ACM module accepts a certificate chain, verifies the validity of each certificate and extracts the attribute information bound with the requestor's identity.

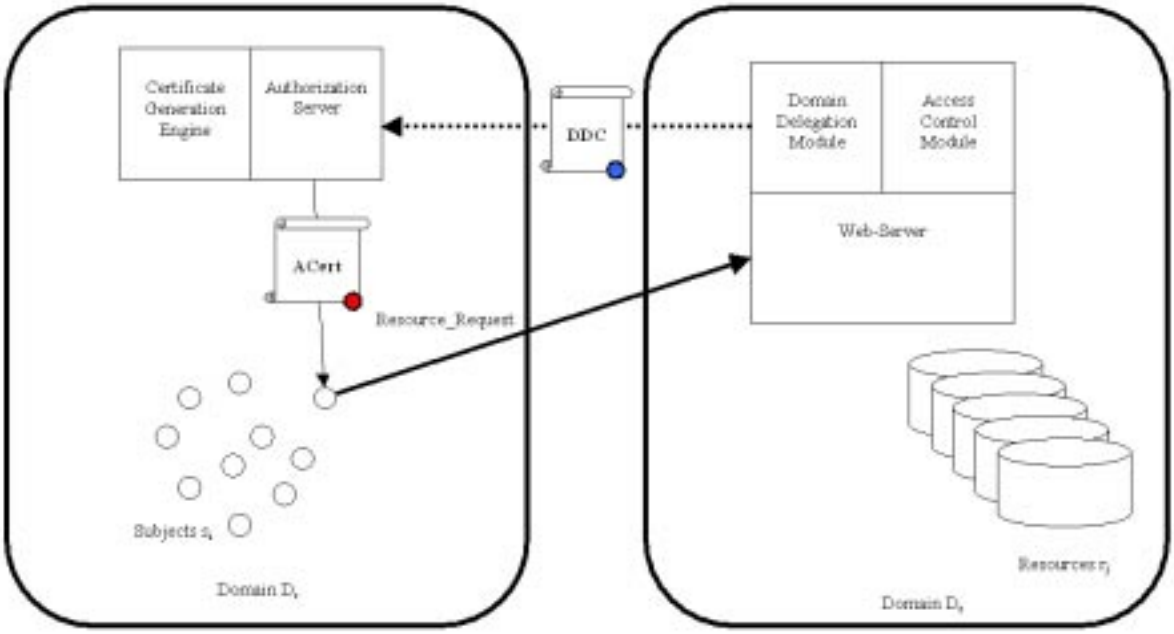

Figure 1. Inter-domain RBAC scenario.

The authority controlling domain D_r (A_r) uses the DDM module to grant the authority controlling domain D_s (A_s) the rights to issue attribute certificates for subjects in domain D_s. Such an initial agreement between the two domain authorities is a prerequisite to be able to issue further inter-domain attribute certificates. Furthermore, the domain delegation

module is used to define the attributes that can be used in the communications between the two partner domains. In case of RBAC, as with the proposed solution, the DDM module is used to define the possible roles that can be granted to users. DDM module also controls the role-to-permission mapping required by the access control decisions performed by the ACM module.

The client domain includes the Authorization Server (AS) that is managed by the domain authority and integrated with a Certificate Generation Engine that is used to generate the attribute certificates for the users. The details on the certificate types will be given in section 2.2. Using the AS the authority A_s can establish an identity-to-role relation: the identity information is provided through the user authentication mechanism performed by the SSLv3 protocol. On the other hand, from the point of view of the user, the integration of the access control functionality is performed through signed applets that are used for an attribute certificate request and for an ordinary resource request.

2.2. Certificates

When a user from domain D_s requests an object stored in domain D_r, the request must be completed with the attribute certificate and the DDC. Furthermore, in order to obtain the attribute certificate, the user has to follow the client authentication process supported by the SSLv3 protocol. Three different certificates are used to complete a transaction: the X509v3 public-key certificate, the attribute certificate and the domain delegation certificate.

- *Public-key Certificate:* an X.509 certificate is used to bind a public-key to a particular individual or entity, and it is digitally signed by the issuer of the certificate (certificate authority) that has confirmed the binding of the public key to the holder (subject) of the certificate [8].

- *Attribute Certificate:* the attribute certificate (ACert) is a digitally signed data structure stating that a subject has a particular attribute [7, 9]. In the case of the RBAC architecture, the attribute is a role. The solution proposed in this paper is based on the use of SPKI authorization certificates. Conceptually, a SPKI authorization certificate consists of five fields that have security relevance, and a signature. More formally, this kind of certificate may be expressed as a digitally signed tuple (I, S, D, A, V), where I and S respectively are the public key of the issuer and the public key of the subject. Instead of the public key itself I and S can alternatively take on the value of the hash of the corresponding public key. The other fields respectively are the delegation bit, the attribute field and the validity period.

- *Domain Delegation Certificate:* the domain delegation certificate (DDC) is a SPKI authorization certificate that authorizes the client domain authority to issue certificates on the behalf of the server domain authority.

2.3. Binding Identities to Attributes

Access control decisions made in the server domain use the ACert provided with the request to grant or deny access to resources [2].

The mechanism proposed in this paper allows the integration of SPKI certificates at the application layer and relies on the underlying SSLv3 protocol. The idea is that if it is possible to securely link the SPKI certificate to an X509v3 certificate, then the challenge-response implemented for the SSL client authentication protocol can be used to assure the

correspondence between the user and the attribute certificate. The binding between an X509 identity certificate and an SPKI authorization certificate is depicted in Figure 2.

If the value of the field corresponding to the Subject of the SPKI authorization certificate is the hash of the public-key stored in the X509 certificate then the authentication protocol not only proves the client's identity, but it also allows the authorization protocol to verify that the attribute certificate was grantedto that particular user. Indeed, if the hash of the public-key related to the current SSL session is equal to the one provided with the ACert, then it is possible to affirm that the client of the SSL session has the attribute listed in the SPKI authorization certificate.

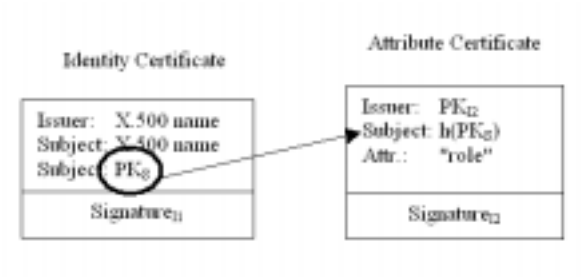

Figure 2. Certificate Types and secure binding.

3. **Description of protocol steps**

From an operational point of view, a typical inter-domain transaction involving subjects (s_i) from domain D_s and objects or resources (r_j) from domain D_r consist of four different phases. The set-up phase, the role-to-permission definition phase, the identity-to-role definition phase and the actual inter-domain transaction.

The set-up phase involves the authority A_r and the authority A_s: an initial agreement between the two domains has to be defined in order to grant the authority A_s the permission to issue attribute certificates for the subjects s_i in its domain. The set-up phase is completed when A_r issues a DDC certificate to A_s, which is a necessary condition in order to complete any inter-domain transaction. Furthermore, A_r and A_s must agree on a definition of a set of roles that can be granted to the subjects that need to access the resources.

The next step consists in the definition of the role-to-permission mapping. This phase take place in the resource domain: the authority A_r associates for each possible role a set of access permissions to the resources. The role-to-permission mapping is necessary for the ACM module in order to take the decision to grant or to deny the access to the requested resource.

The identity-to-role definition phase establishes the mapping between the identity of a subject s_i and the role he or she has in domain D_s. The identity-to-role mapping takes the form of an attribute certificate that is issued and signed by the authority A_s. Subjects have to follow the client authentication process supported by the SSLv3 protocol to complete an ACert request. Furthermore, thanks to the mechanism used to securely bind identities and roles (see section 2.3), the authority A_s can be sure that the ACert is issued for the correct subject and that it can be used only by that subject.

The inter-domain transaction can only take place when the previous phases are successfully completed. An inter-domain communication involves a subject s_i belonging to domain D_s and resources r_j belonging to domain D_r. s_i builds a resource request and sends it to the D_r web server which elaborates the request and grants or denies access to the requested resource. The resource request is sent over an SSLv3 channel with mutual

authentication (for both the client and the server) and is composed of three[1] different fields: the requested object r_j, the requestor's ACert and the DDC. The DDC is necessary for the ACM module to verify the prior agreement between the two communicating domains. The ACert is used to extract role information about the requestor. Furthermore, since the request is sent over an authenticated SSL session, the ACM module can verify the binding between the requestor identity and the presented ACert (see Figure 2). Finally, the ACM module uses the role-to-permission mapping to decide whether to grant or deny the access to the resource.

4. Application Scenario

The security platform that was implemented as part of the European Project ESW has been tested for a payroll application. The typical scenario involves a company (Client Company) that needs to outsource the payroll management process to a service provider(Pay Service). Using the same notation as in section 3, the domain of Client Company corresponds to the subject domain D_s while the Pay Service domain corresponds to the resource domain D_r. The subjects s_i are the Client Company employees while the resources r_j are the payroll information of each emloyee.

During the set-up phase the Pay Service authority (A_r) and the Client Company authority (A_s) define a set of roles that can be credited to users in the subject domain: an example set could be {director, accountant, manager, engineer}. A_r then defines the role-to-permission mapping: an example is depicted in the following Table 1.

Role	**Permission**
Director	[Read], all payroll information
Accountant	[Read, write, edit], all payroll information
Manager	[Read], team payroll information
Engineer	[Read], personal payroll information

Table 1. Role-to-permission mappings.

On the other side, A_s prepares and issues an attribute certificates for each subject of its domain.

The advantage of an AC architecture based on the RBAC model is considerable when the number of subjects s_i is higher than the cardinality of the role set. Indeed, compared to a generic AC model, A_r has to manage the mapping table between a small set of roles and permissions instead of a large number of subjects and permissions. Furthermore, the binding between the subject identity and the possible roles is left at the discretion of an authority (A_s) responsible for the subject domain.

The security platform based on the original mechanism proposed in this paper (see section 2) inherits all the advantages of a RBAC model and can be seamlessly integrated in actual client-server applications.

[1] When the role delegation property is used, the resource request might be composed of more than three fields, but the delegation feature of our mechanism will be the subject of a further paper.

5. Related technologies and software implementation

Current approaches to perform role based access control on Web servers are mostly based on attribute certificates. It is possible to find definitions and implementations of attribute certificates such as the one depicted in [1] or the one defined in the SESAME project [3].

Smart Certificates [1] are based on the X509v3 standard: both the attributes and public-key information are bundled in a single certificate. The attribute information is stored in the extension field of the X509v3 certificate and can be signed by a certification authority different from the one that signed the public-key certificate.

SESAME implements the ECMA-219 Privilege Attribute Certificate (PAC). In the particular case of a non-delegable PAC, the certificate is bound to an identity: a Privilege Attribute Server (PAS) will issue the PAC certificate only to a user that can prove the possession of a validation key. The SESAME security mechanisms are not implemented nor supported in the security modules bundled with current browsers making a seamless integration a difficult task.

Compliance with X509v3 standard is the key requirement for the integration of new services into the existing security packages used in the Internet environment.

Despite their compliance with the X509v3 standard, Smart Certificates are inappropriate for time-variant attribute-identity mappings as required by RBAC. Smart Certificates append attribute certificate information to the content of X509v3 identity certificates. Thus, every update of the attribute information requires the generation of a new certificate, that is, the computation of a digital signature by the certification authority.

The key feature of the security platform presented in this paper is a seamless integration with existing client-server software.

The core modules of the proposed architecture (the AS module, the DDM module and the ACM module, see Figure 1) are Java-based servlets that can be integrated as plug-ins in web-server software that support servlets technology. On the client side, the management of attribute certificates and the construction of a resource request (see section 3) are achieved by signed Java applets that are supported by most existing browsers. Furthermore, the attribute certificate mechanism relies on the existing SSLv3 protocol, which is a de-facto standard for secure communications (see section 2.3). Therefore, our RBAC platform can be integrated in existing web applications without major architectural modifications.

6. Conclusions and future work

Attribute certificates appear as the key requirement for a meaningful application of existing PKI based security services in the area of business transactions. The approach presented in this paper demonstrated that practical solutions for the inter-domain access control problems can be achieved in a transparent manner using existing technologies. This solution was implemented and is currently being optimized for a European R&D project. Furthermore, trial tests are envisaged to study the impact of such security platform on business transactions performed by actual e-commerce applications. Further studies are envisaged to propose the integration of the presented authorization service in the SSLv3/TLS protocol.

References

[1] J. S. Park, R. Sandhu, Smart Certificates: Extending X.509 for secure attribute services on the web.
[2] J. S. Park, R. Sandhu, Binding identities and attributes using digitally signed certificates.
[3] M. Vandenwauver, R. Govaerts, J. Vandewalle, How Role Based Access Control is implemented in SESAME
[4] J. S. Park, R. Sandhu, Decentralized User-Role Assignment for Web-based Intranets
[5] R. Sandhu, Role Based Access Control
[6] D. Ferraiolo, J. Cugini, D. Kuhn, Role-Based Access Control (RBAC): Features and Motivations
[7] W. Johnston, S. Mudumbai, M. Thompson, Authorization and Attribute Certificates for Widely Distributed Access Control
[8] ITU-T X.509
[9] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, SPKI Certificate Theory - RFC 2693