ROLE-BASED EAM USING X.509 ATTRIBUTE CERTIFICATE[*]

Dongwan Shin and Gail-Joon Ahn
Department of Software and Information Systems
The University of North Carolina at Charlotte
{doshin, gahn}@uncc.edu

Sangrae Cho
Department of Information Security System
Electronics and Telecommunications Research Institute
sangrae@etri.re.kr

Abstract In this paper, we describe an experiment of designing and implementing a role-based extranet access management (EAM) by leveraging role-based access control (RBAC) and X.509 attribute certificate for scalable and interoperable authorization. Compared with previous works in this area, we show that our approach can overcome the problems of previous solutions and broaden RBAC's applicability into large-scale networks. The components for role administration are defined and a security architecture is discussed. We also demonstrate the feasibility of our approach through a proof-of-concept implementation. Several issues from our experiment are briefly discussed as well.

Keywords: Access control, Role-based, Attribute certificate, Privilege management infrastructure

1. Introduction

Extranet access management (EAM) has received much attention in recent years as a solution of security challenges that web-based applications are faced with. EAM is often referred to as a unified mechanism for both managing the authentication of users across enterprises (i.e., single

[*] Research supported at the Laboratory of Information of Integration, Security and Privacy at the University of North Carolina at Charlotte by grants from the Electronics and Telecommunications Research Institute.

sign-on) and implementing business rules for determining user access to enterprise applications and their resources. These business rules are utilized for authorization in a security context [14]. Authentication mechanisms for EAM have been practiced at considerable length and various authentication schemes such as SSL, LDAP-based, or secure cookies-based have been widely accepted. Unlike authentication mechanisms in EAM, authorization mechanisms in EAM which can conveniently enforce various business rules from different authorization domains among various applications still need to be investigated.

Role-based access control (RBAC) has been acclaimed and proven to be a simple, flexible, and convenient way of managing access control [1–2]. In RBAC, access control depends upon the roles of which a user is a member, and permissions are assigned to the roles. This extremely simplifies management of permissions, reducing complexity and potential errors in directly assigning permissions to users. RBAC is also flexible enough to satisfy different organizational access control policies such as least privilege, separation of duties, and abstract operations. This flexibility is beneficial to organizations that need to modify their access control policies for their needs.

As a major component in privilege management infrastructure (PMI) [10–11], X.509 attribute certificates allow us to construct a scalable and interoperable authorization infrastructure. PMI is composed of various components such as attribute certificates, attribute authorities, repositories, entities such as privilege asserters and verifiers, objects, and object methods. It provides certificate-based authorization with attribute certificates while public-key infrastructure (PKI) does certificate-based authentication with public-key certificates, so called identity certificates. The main difference between these two certificates is that attribute certificates do not contain public key, whereas public-key certificates do. Attribute certificates bind entities to attributes, which may be the entities' role or group information, while public-key certificates bind entities to public key. One of the great benefits of PMI is to establish the trustworthiness among different authorization domains as long as each of them keeps the meaning of attributes intact. Thus, access control could be enforced not just within a single authorization domain, but also across multiple domains.

Our objective in this paper is to design a role-based EAM leveraging RBAC features and X.509 attribute certificate. We attempt to develop an easy-to-use, flexible, and interoperable authorization mechanism for EAM. We identify a set of components that is necessary to pursue our goal and develop an appropriate system architecture. Also, we demonstrate the feasibility of our architecture by providing the proof-of-concept prototype implementation using commercial off-the-shelf (COTS) technologies.

The rest of this paper is organized as follows. Section 2 shows previous researches related to our work. Section 3 gives an overview of background technologies. Section 4 describes our approach to designing a role-based EAM and its system architecture. Implementation details are described in section 5, including a case study with a hypothetical enterprise. Section 6 discusses lessons learned from our experiment and concludes the paper.

2. Related Works

Several researchers have been trying to accommodate RBAC features into large-scale systems of intranet or extranet focusing on various applications such as database systems, web servers, or web-based workflow systems. There have been also rigorous researches on how to use attribute certificates for the purpose of managing privileges on distributed computing systems.

In the OSF/DCE environment [12–13], privilege attribute certificate (PAC) that a client can present to an application server for authorization was introduced. PAC provided by a DCE security server contains the principal and associated attribute lists, which are group memberships. The application server works as a reference monitor to make access control decisions based on the comparison between the client's attributes and attributes in ACLs. This approach focused on the traditional group-based access control.

Similarly, Thompson et al. [9] developed a certificate-based authorization system called Akenti for managing widely distributed resources. It was especially designed for system environments where resources have multiple stakeholders and each stakeholder wants to impose conditions for access. There are two types of certificates employed for authorization: use-condition certificate and attribute certificate. The stakeholders assert their access requirements in use-condition certificates and an attribute authority issues attribute certificates that bind a user to attributes. Their approach emphasized the policy-based access control in a distributed environment.

Also, several studies have been carried out to make use of RBAC features with the help of public-key certificates [5–7]. Public-key certificates were used to contain attribute information such as role in their extension field. Two architectures have been identified in [7]: user-pull and server-pull. [6] demonstrated how RBAC can be injected to secure a web-based workflow system using the user-pull style architecture

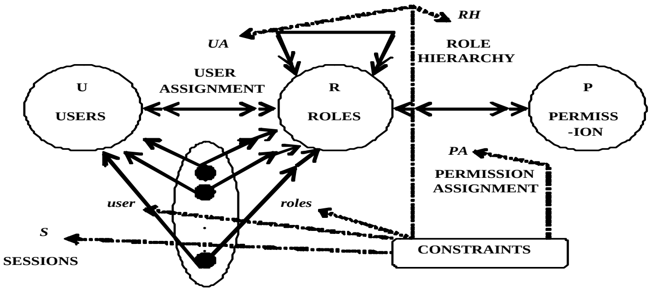


Figure 1. RBAC$_3$ Model.

whereas [5] described the server-pull style architecture for role-based authorization, adopting an LDAP-oriented approach. To add role information into public key certificates, however, may cause problems such as shortening of certificates' lifetime and complexity of their management. A user's role memberships are dynamic entities even though roles themselves are persistent, compared to the user's identity. Whenever role memberships change, a new public key certificate binding the user's identity and new roles needs to be issued. Subsequently, it leads unnecessary revocation of a public-key certificate which could be still valid for identity affirmation purposes.

3. Background Technologies

3.1 Role-based access control

RBAC begins with the basic concept of roles. Roles are defined based on job functions or job titles within an organization. Permissions are assigned to the roles, and then users are associated with appropriate roles. A well-defined family of RBAC models was presented in [1]. The four conceptual models discussed are RBAC$_0$, RBAC$_1$, RBAC$_2$, and RBAC$_3$. RBAC$_0$ is the base model, made up of six components. Those components are expressed in formal representation below from *i* through *vi* [1]:

i. U is a set of users,
ii. R is a set of roles,
iii. P is a set of permissions,
iv. S is a set of sessions,
v. UA $\subseteq$ U $\times$ R, is a many-to-many user to role assignment relations,
vi. PA $\subseteq$ P $\times$ R, is a many-to-many permission to role assignment relations, and
vii. RH $\subseteq$ R $\times$ R, is partially ordered role hierarchies.

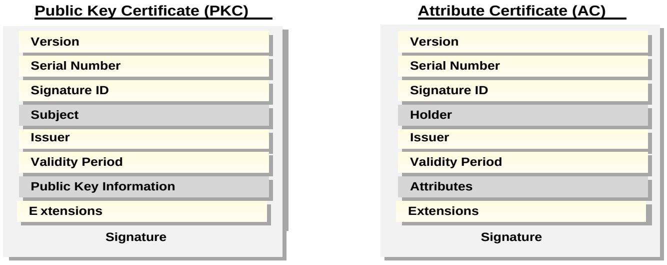


Figure 2. Difference between public key certificate and attribute certificate.

RBAC₁ and RBAC₂ introduce advanced constructs to the base model with notions of role-hierarchies and constraints, respectively. RBAC₃, as a consolidated model, combines RBAC₁ and RBAC₂ as shown in Figure 1. Role hierarchies [1, 4], as an added feature in RBAC₁, are an important construct to build the hierarchical relationship between roles. They represent the lines of authorities and responsibilities within an organization. They also help role management be flexible and simple. Senior roles inherit the permission assigned to junior roles. Formal notation of role hierarchies is also expressed above (see *vii*). Constraints are an essential construct needed for laying out higher-level access control policies within an organization [3]. They can be applied to any of the above-mentioned components as shown in Figure 1. A well-known example is separation of duty that reduces possible frauds or errors by controlling membership in, activation of, and use of roles as well as permission assignment.

3.2 Privilege Management Infrastructure

PMI is based on the ITU-T Recommendation of directory systems specification [10], which introduced PKI in its earlier version. As we discussed in Section 1, public-key certificates are used in PKI while attribute certificates are a central notion of PMI. Public-key certificates are signed and issued by certification authority (CA), while attribute certificates are signed and issued by attribute authority (AA). PMI is to develop an infrastructure for access control management based on attribute certificate framework. Attribute certificates bind attributes to an entity. The types of attributes that can be bound are role, group, clearance, audit identity, and so on. Attribute certificates have a separate structure from that of public key certificates. Figure 2 shows the difference between public key certificate and attribute certificate.

PMI consists of four models: general model, control model, delegation model, and roles model. General and control models are required, whereas roles and delegation models are optional. The general model provides the basic entities which recur in other models. It consists of three foundation entities: the object, the privilege asserter, and the privilege verifier.

◇ *Object*: a resource being protected.
◇ *Privilege asserter*: the entity that holds a particular privilege and asserts its privileges.
◇ *Privilege verifier*: the entity that makes the determination of pass/fail on usage of object.

The control model explains how access control is managed when privilege asserters request services on object. The following three entities are added in the control model.

◇ *Privilege policy*: the degree of privilege used in the determination of pass/fail on usage of object.
◇ *Environmental variable*: the local aspect of policy such as time and date.
◇ *Object method*: the attributes of the request such as read and execute.

When the privilege asserter requests services by presenting his/her privileges, the privilege verifier makes access control decisions based upon the privilege presented, privilege policies, environmental variables, and object methods.

The delegation model handles a situation when privilege delegation is necessary. It introduces two additional components: source of authority (SOA) and other attribute authorities (AAs). When delegation is used, SOA assigns privilege to AAs, and AAs delegate privileges to an end-entity privilege asserter. Lastly, PMI roles model also introduces two additional components: role assignment and role specification. Role assignment is to associate privilege asserters with roles, and its binding information is contained in attribute certificate called role assignment attribute certificate. The latter is to associate roles with privileges, and it can be contained in attribute certificate called role specification attribute certificate or locally configured at a privilege verifier's system.

4. Role-based EAM Using Attribute Certificate

Our approach is based upon PMI roles model. Accordingly, two different attribute certificates are employed: role assignment attribute certificate ($RAAC$) and role specification attribute certificate ($RSAC$). The integrity of the bindings is guaranteed through digital signature in attribute certificate. Figure 3 shows the structures of two attribute certificates.

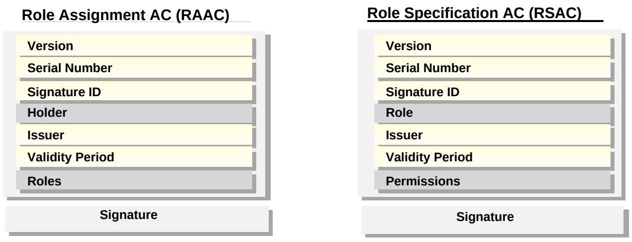


Figure 3. Role assignment and role specification attribute certificate.

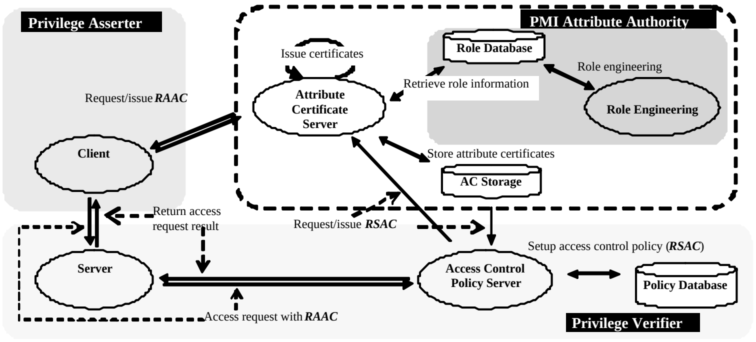


Figure 4. Operational architecture for role-based EAM.

In addition, we need to design the system architecture that can fully utilize these attribute certificates. Our architecture consists of three components. They are privilege asserter, privilege verifier, and PMI attribute authority, as shown in Figure 4. Privilege asserter is a client. The client is a user or a system. It asks for and retrieves $RAAC$s from PMI attribute authority and requests access to web services (or protected resources). Note that we omit authentication procedures in our architecture for brevity.

Privilege verifier is composed of server, access control policy server, and policy database. The server could be a resource server or an application server. When a client wants to access the server, the server asks the access control policy server if the client has the access privilege. The access control policy server makes access control decisions based on both the client's roles from a $RAAC$ and the permissions assigned to the roles from a $RSAC$. The $RSAC$ can be obtained from the PMI attribute authority or the policy database. The policy database maintains all *RSAC*s that are previously retrieved from the PMI attribute authority.

PMI attribute authority has four entities: attribute certificate server, AC storage, role database, and role engineering administration. The attribute certificate server signs and issues both *RAAC*s and *RSAC*s. After issuing those certificates, it stores them into a publicly accessible repository, AC storage. Private role database retains RBAC components enabling a role-based authorization infrastructure. RBAC components are built through role engineering administration.

5. Implementation Details

5.1 Role-based EAM

Our implementation tasks are divided into two phases. The first phase is to build APIs for both a role-based decision making engine and attribute certificates. Those APIs are the core building blocks for constructing an access control policy server and an attribute certificate server. The second phase is to implement each entity integrating with APIs. Currently we are in the transition period from the first phase to the second.

5.1.1 Privilege asserter. We developed a privilege asserter using ActiveX control, named attribute certificate manager. The manager enables a user to import downloaded BER-encoded RAACs into Windows registry. It also allows the user to view and select one of RAACs in the registry. The selected RAAC will be presented for requesting access to resources.

5.1.2 Privilege verifier. Internet Information Server (Version 5.0) is used as a server. An HTTP raw data filter, called AC filter, was developed using Microsoft ISAPI (Internet Server API) technology. Its main task is screening the incoming raw data from a client to see if the client presents any attribute certificate. We also developed an application working as an access control policy server. An engine for making access control decisions is a major component in this application. Figure 5 shows the procedures of making access control decisions in this engine.

5.1.3 PMI attribute authority. An attribute certificate server was developed to generate *RAAC*s and *RSAC*s. The programming library, called AC SDK, was built for supporting the functionality related to the generation of the attribute certificates. Netscape Directory Ser-

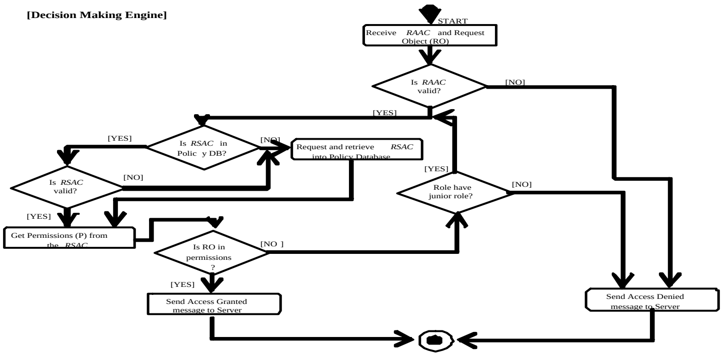

[Decision Making Engine]
START
Receive RAAC and Request Object (RO)
Is RAAC valid?
[NO]
[YES]
Is RSAC in Polic y DB?
[YES]
[NO]
Request and retrieve RSAC into Policy Database
Is RSAC valid?
[NO]
[YES]
Get Permissions (P) from the RSAC
Is RO in permissions ?
[NO]
[YES]
Role have junior role?
[YES]
[NO]
Send Access Granted message to Server
Send Access Denied message to Server

Figure 5. Activity diagram of role-based decision-making engine.

vice 5.0 was used for both a role database and an AC storage. For the purpose of designing and implementing the role engineering administration, we identified three categorical components: structural, functional, and informational. The structural component consists of sets of entities in RBAC models: roles, role-hierarchies, users, permissions, objects, operations, and constraints. The functional component enables the relationship among the entities. The informational component represents repositories such as the LDAP directory service server and it contains a database for role-hierarchy information, a database for role-permission information, and a database for role-user information. We implemented a Java-based stand-alone application for the role engineering administration.

5.2 A Case Study

In order to demonstrate the feasibility of our approach, we developed a hypothetical enterprise for our case study. Suppose we have an online brokerage A and this company provides three different web-based services: simple quote service, stock trading, and online banking service. Suppose also the online banking service is provided as a result of the recent coalition with an online bank B. Both companies leverage PKI for the security of transactions as well as an authentication service. In addition, they use a role-based EAM system for an authorization service. Online brokerage A's customers are assigned into three different roles: guest, user, and power user. power user is the most senior role, guest role is the most junior role, and user role is in between them. Customers having guest role can only get the permission for the simple quote service,

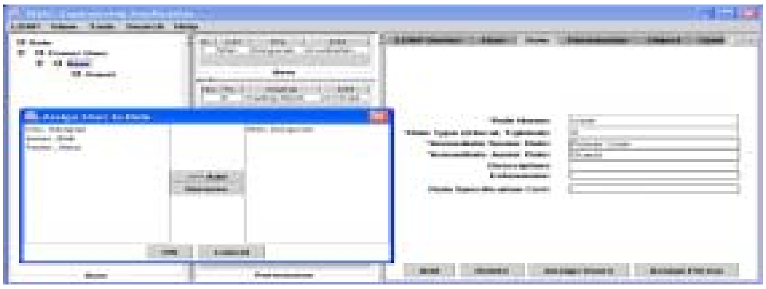


Figure 6. Role engineering administration in user-to-role assignment.

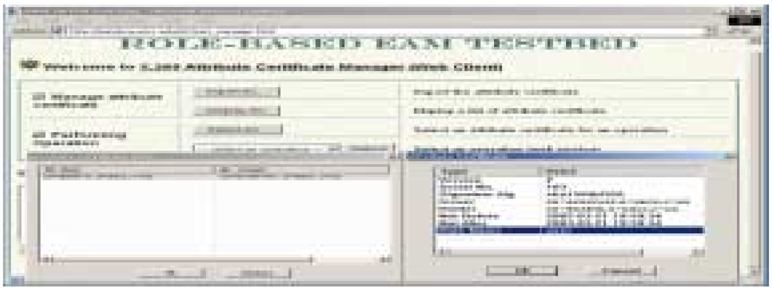
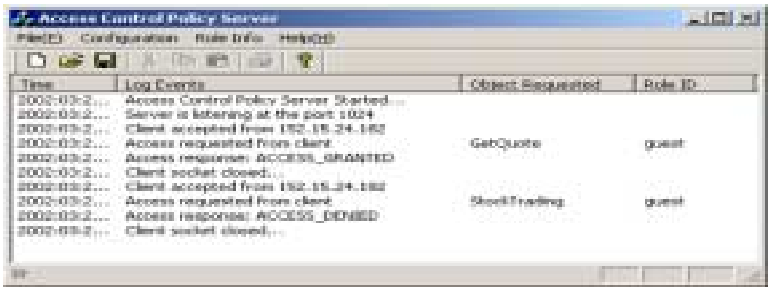


Figure 7. X.509 attribute certificate manager.



```
Time           Log Events                                    Object Requested    Role ID
2002-03-2...   Access Control Policy Server Started...
2002-03-2...   Server is listening at the port 1024
2002-03-2...   Client accepted from 152.15.24.182
2002-03-2...   Access requested from client                  GetQuote            guest
2002-03-2...   Access response: ACCESS_GRANTED
2002-03-2...   Client socket closed...
2002-03-2...   Client accepted from 152.15.24.182
2002-03-2...   Access requested from client                  StockTrading        guest
2002-03-2...   Access response: ACCESS_DENIED
2002-03-2...   Client socket closed...
```

Figure 8. Access control policy server.

user role customers are given the permissions for trading stocks as well as the quote service. power user role customers can have the permissions for all three services.

User-to-role and permission-to-role assignments are done through the role engineering administration we described in 5.1.3. Figure 6 shows its interface. It illustrates that Dongwan is the current member of user role and three users on the left can be assigned to the role (when a role in the leftmost pane is clicked, the application shows all information related to the role, i.e., the current members, permissions, and the properties).

With the attribute certificate manager, customers can easily manage their attribute certificates and use them for authorization. Figure 7 shows its interface. The small window in the right illustrates the content of the retrieved $RAAC$, while the small window on the left shows the imported attribute certificates. By submitting the $RAAC$ to the online brokerage A's web server using the attribute certificate manager, customers can request the service that he wants. Customers with the power user role can get the online banking service, because the power user role can be understood and verified by the online bank B's access control policy server. This can be done by utilizing $RSAC$s and the access control policies between A and B. Figure 8 displays the history of activities on the access control policy server.

6. Discussion and Conclusion

The administrative complexities and interoperability are major problems that the current solutions for EAM has been suffering from. In this paper we described an experimental approach to designing a role-based EAM leveraging RBAC and X.509 attribute certificate. The system architecture was discussed and the components necessary for engineering and administration of roles were also defined. In addition, we demonstrated the feasibility of our approach through a proof-of-concept implementation. Compared with others works, our approach could overcome problems such as undesirable shortening of public key certificate's lifetime when the extension field of public-key certificates is used to contain role information. Also our work confirmed that RBAC could be applied to broader scale environments.

An issue lingering throughout our implementation was an alternative to ASN.1 encoding for role-related security credentials in attribute certificates used in PMI. XML has become the standard for data exchange on the web and is more intuitive, compared with ASN notations. Security Assertion Markup Language (SAML) is an XML-based framework for exchanging security credentials such as role information. We will investigate how XML-based credential encapsulation can be used in our architecture. And we are just beginning to understand the implications of large-scale and heterogeneous PMIs in this paper. Another direction that we have to consider is to illustrate the semantic meanings of attribute certificate revocation scheme and their impact on the process of our approach. Our future study will include a practical solution to provide appropriate models for attribute certificate revocation. The delegation model in PMI was outside of the scope of our work in this paper even though we have previously introduced a role-based delegation framework [8]. We will investigate how our delegation model can be applied in this work.

References

[1] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. “Role Based Access Control Models,” *IEEE Computer* 29 (2), February 1996.

[2] D. Ferraiolo, J. Cugini, and D.R Kuhn. “Role Based Access Control: Features and Motivations,” In *Annual Computer Security Applications Conference*, IEEE Computer Society Press, 1995.

[3] G.Ahn and R. Sandhu. “Role-based Authorization Constraints Specification,” *ACM Transactions on Information and System Security*, 3(4), November 2000.

[4] R. Sandhu. “Role-hierarchies and Constraints for lattice-based access control,” In *Proceedings of 4th Enropean Symposium on Research in Computer Security*, Rome, Italy, 1996.

[5] J. Park, G. Ahn, and R. Sandhu. “RBAC on the Web using LDAP,” In *Proceedings of the 15th IFIP WG 11.3 Working Conference on Database and Application Security*. Ont., Canada, July 2001.

[6] G. Ahn, R. Sandhu, M. Kang, and J. Park. “Injecting RBAC to secure a Web-based workflow system,” In *Proceedings of 5th ACM Workshop on Role-Based Access Control*. Berlin, Germany, July 2000.

[7] J. Park, R. Sandhu, and G. Ahn. “Role-based Access Control on the Web,” *ACM Transactions on Information and System Security*, 4(1), February 2001.

[8] L. Zhang, G. Ahn, and B. Chu. “A Rule-Based Framework for Role-Based Delegation,” In *Proceedings of ACM Symposium on Access Control Models and Technologies*, Chantilly, VA, May 2001.

[9] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari. “Certificate-based Access Control for Widely Distributed Resources,” In *Proceedings of the 8th USENIX Security Symposium*, Washington, D.C., August 1999.

[10] ITU-T Recommendation X.509. Information Technology: Open Systems Interconnection - The Directory: Public-Key And Attribute Certificate Frameworks, 2000. ISO/IEC 9594-8:2001.

[11] S. Farrell and R. Housley. An Internet Attribute Certificate Profile for Authorization, PKIX Working Group, June 2001.

[12] OSF DCE 1.0 Application Development Guide, Open Software Foundation, Cambridge, MA, 1992.

[13] OSF DCE 1.0 Introduction to DCE, Open Software Foundation, Cambridge, MA, 1999.

[14] John Pescatore. Extranet Access Management Magic Quadrant, *Gartner Research Note (ID: M-13-6853)*, Gartner INC., May 2001.