Proceedings of the 2003 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY June 2003

Secure Web-based Applications with XML and RBAC

Cungang Yang and Chang N. Zhang, *IEEE member*

Abstract

In this paper, a Role-based Access Control model for web-based applications (ORBAC) is introduced. Also, an efficient method for managing ORBAC security policies using XML and a role assignment algorithm are presented. Unlike most existing approaches, with our approach the authorization is independently defined and is separated from implementation mechanisms.

INDEX TERMS–DIGITAL CREDENTIAL, ORBAC, RBAC, XML.

1. INTRODUCTION

Nowadays, there is a growing concern over the security of web-based applications which are rapidly being deployed over the Internet. Public-key infrastructure (PKI) has been an important development for addressing the security concerns of web-based applications. Users can be authenticated using Public Key Infrastructure facilities, however, such facilities do not provide access control mechanism.

Current access control approaches on web using username and password to differentiate users is no longer sufficient as knowledge of a user's identity will often not be sufficient to determine whether a user is authorized to a privilege, it is possible that an authorized user on web is non-predefined and we have less prior knowledge about the users on web.

In order to deal with the access control problem on web, it is required to establish the relationships between users in Internet environment and some trustworthy needs to be established. Since digital credentials can be used to manage trust establishment more efficiently, the research goal is therefore to explore the use of digital credential on the security of web-based applications.

An implementation of a RBAC system [1][2][3] for the web environment has been reported by Ferraiolo, D.F., Barkley, J.F., and Kuhn, D.R [4]. The implementation consists of a web server to enforce RBAC and an administrative tool to allow security administration. But the system places no requirements or security policy for the users on the browser. When a user issues an access request, a role is assigned to the requester after establishing a session using the available authentication and confidentiality services.

Nanthan N. Vuong and Geoffrey S. Smith [5] describes practical concepts that can be employed in a single domain for managing security policies using XML, but it has not extended the security policies for Internet users.

The rest of the paper is organized as follows. Section 2 introduces the Role-based Access Control model for web-based applications (ORBAC). Section 3 presents the XML-based ORBAC security policy, a role assignment algorithm is also presented. Finally, section 4 concludes the paper.

2. Role-Based Access Control Model for Web-based Applications

I proposed a role-based access control model for web-based applications that extends the original Role-based Access Control from single enterprise domain to web-based implementations. In the Internet environment, trust establishment may be applicable wherever Internet users want to engage in sensitive transactions without sufficient pre-established trust, such applications involve essentially every aspect of e-commerce. For instance, a healthcare enterprise's policy could enable access to anonymous medical files for research purpose but limits only to authorized Internet users, the healthcare enterprise may allow some medical files can be access by medical doctors, some medical files can be accessed by registered nurses. A *privilege* in the model is an access mode that can be exercised on objects, for instance, reading the abstract of a medical file or buying a medical file on Internet are privileges that can be assigned to the Internet users. Role in the model is a collection of privileges performed by a group of Internet users who are members of certain digital credentials, for instance, a role could be authorized to a group of Internet users who are members of medical doctor credential and another role could be authorized to a group of Internet user who are members of registered nurse credential, etc. A role may have multiple privileges, and the same privilege can be associated to different roles. In the proposed model, Internet users do not directly associated with the privileges; instead, these privileges are associated with roles, Internet user is assigned to one or multiple members of roles and each role is assigned to one or multiple Internet users. In the model, a *role hierarchy* is introduced to reflect inheritance of authority and responsibility among the roles. These ideas greatly simplify the management of authorizations and reduce the management costs. The model is shown in Fig. 1 and the class diagram of the model is shown in Fig. 2.

ISBN 0-7803-7808-3/03/$17.00

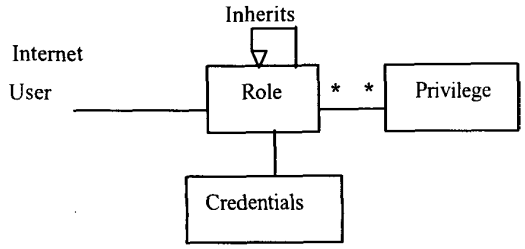

Fig. 1 Role-based Access control Model for Web-based applications (ORBAC)

2.1 Credential

A credential contains values for all the subject properties specified in a credential type. There are multiple subject properties and subject property values for a credential. For instance, subject properties of a medical doctor credential could be profession, Issuer, etc. The subject property value of "profession" could be "medical Doctor" and the subject property value of "Issuer" could be "VeriSign". Credentials in the proposed model are used to determine whether or not roles can be authorized to Internet users. For instance, if a security policy of the healthcare enterprise defines that a role H: Medical doctor Credential (Profession="Medical Doctor", Issuer= "Verisign", etc) only can be authorized to internet users who are member of medical doctor credential and satisfy the subject property and subject property values defined in H, then for the Internet users who want to be a member of the role H, they must provide a medical doctor credential first. Secondly, in the submitted medical doctor credential, subject property profession must be medical doctor, the subject property issuer must be "Verisign", otherwise, their applications will be rejected.

Sometimes, Internet users may provide more than one credentials to be a member of a role. For instance, in order to buy a medical file on the Internet, Internet users could be required to provide a medical doctor credential and a Visa card credentials or a medical doctor credential and a Master card credentials. Therefore, a credential chain will be established. *A credential chain* contains multiple credentials which linked by logical AND "$\wedge$". Credentials linked with logical OR "$\vee$" belong to different credential chains. For instance, a logical expression of credentials for a role H, (credential C6$\wedge$credential C5)$\vee$(credential C6$\wedge$credential C7), defines two credential chains (C6$\wedge$C5) and (C6$\wedge$C7).

C6: Medical Doctor Credential (Profession=" Medical Doctor", Issuer="VeriSign", etc.)

C5: Master Card Credential (Expiration date> "01/08/2005", Issuer="CIBC", etc.)

C7: Visa Card Credential (Expiration date> "05/06/2005", Issuer="CIBC", etc.)

If Internet users would like to be a member of role H, a medical doctor credential and a Master card credential or a medical doctor credential and a Visa card credential should be submitted. Also, the submitted credentials must satisfy the subject properties and subject property values defined in credential C6 and credential C5 or credential C6 and credential C7. In any credential chain, we define that a credential is *a parent credential* of the credentials located at right side of it, a credential is a *child credential* of the credentials located at left side of it.

In Fig. 2, credential class has a *credential function* that determine whether or not the credential Internet user submitted satisfies the subject properties and subject property values defined in the credential object.

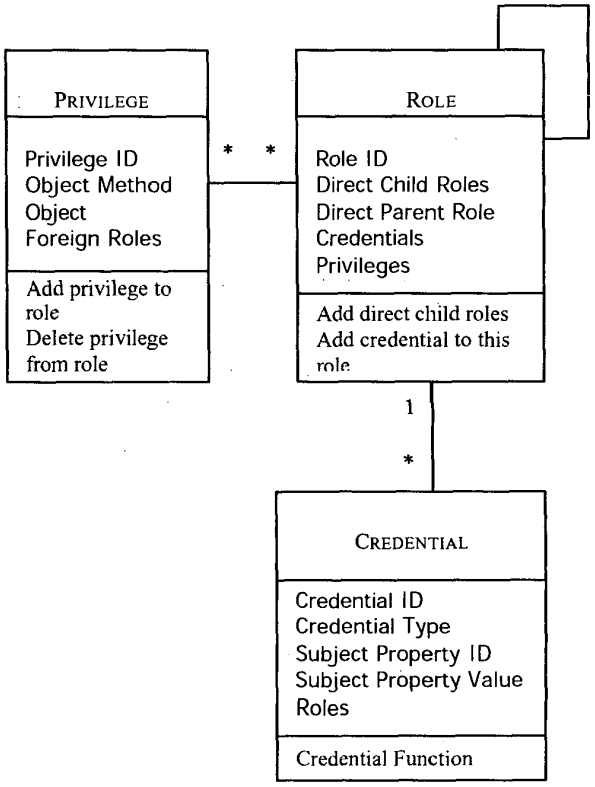

Fig 2. Class diagram of ORBAC

3. XML-based ORBAC Security Policy

XML specification is the work of the World Wide Web Consortium (W3C) Standard Generalized Markup Language (SGML) Working Group [6]. XML precisely and effectively represents desired security policies and offers an additional degree of flexibility. The advantages of using XML are that as a meta language, XML can well define ORBAC security policies and is able to extend and modify them easily. The driving motivation of using XML-based ORBAC security policy is to simplify security policy administration for web-based applications.

ISBN 0-7803-7808-3/03/$17.00 © 2003 IEEE

Authorization is independently defined and is separated from implementation mechanisms.

Each component of the XML-based RBAC security policy is represented by an XML element and the security policy is comprised of the following two parts:
Part 1 defines the syntax of basic elements of the XML-based ORBAC security policy: It includes Privilege, Role and Credential. Part 2 defines the syntax of role hierarchy, privilege assignment and credential assignment. An example security policy is shown as follows.

Example 1: (XML-based ORBAC Security Policy)

```
<? xml version= "1.0" >
<ORBAC-MODEL TYPE= "RBAC1_POLICY">
  <! -- Basic Elements -- >
    <! --Privilege set definition-- >
      <PRIVILEGE ID= "p1" > </PRIVILEGE>
      <PRIVILEGE ID= "p2" > </PRIVILEGE>
      <PRIVILEGE ID= "p3" > </PRIVILEGE>
      <PRIVILEGE ID= "p4" > </PRIVILEGE>
    </ --Privilege set definition-- >
    <! –Foreign Role definition-- >
      < ROLE ID= " J " > </ ROLE>
      < ROLE ID= " H " > </ ROLE>
      < ROLE ID= " I " > </ ROLE>
    </ – Foreign Role definition-- >
    <! --Credential set definition-- >
      <CREDENTIAL ID="C1" TYPE= "Health
Care Provider" >
        <SUBJECT -PPROPERTY ID=
"Profession" OPERATOR= "=" Value= "Health Care
Provider ">
        </ SUBJECT PROPERTY>
        <SUBJECT -PPROPERTY ID= "Valid
Date" OPERATOR= "<" Value= "05/10/2000">
        </ SUBJECT -PROPERTY>
      </CREDENTIAL>
      <CREDENTIAL ID="C2" TYPE= " Nurse">
        < SUBJECT -PPROPERTY ID=
"Profession" OPERATOR= "=" Value= "Nurse">
        </ SUBJECT -PROPERTY>
        < SUBJECT -PPROPERTY ID= "Valid
Date" OPERATOR= "<" Value= "10/10/2001">
        </ SUBJECT -PROPERTY>
      </CREDENTIAL>
      <CREDENTIAL ID="C3", TYPE=" MASTER
Card" >
        < SUBJECT -PPROPERTY ID=
"Expiration Date" OPERATOR= ">" Value=
"06/20/2002">
        </ SUBJECT-PROPERTY>
        < SUBJECT -PPROPERTY ID= "Credit
Value" OPERATOR= ">" Value= "1000">
        </ SUBJECT -PROPERTY>
      </CREDENTIAL>
      <CREDENTIAL ID="C4" TYPE= " VISA
Card">
        < SUBJECT-PPROPERTY ID= "Expiration
Date" OPERATOR= ">" Value= "02/20/2001">
        </ SUBJECT -PROPERTY>
        < SUBJECT-PPROPERTY ID= "Credit Value"
OPERATOR= ">" Value= "5000">
        </ SUBJECT -PROPERTY>
      </CREDENTIAL>
      <CREDENTIAL ID="C5" TYPE= "MASTER
Card">
        < SUBJECT -PPROPERTY ID= "Exp iration
Date" OPERATOR= ">" Value= "07/26/2000">
        </ SUBJECT -PROPERTY>
        < SUBJECT -PPROPERTY ID= "Credit
Value" OPERATOR= ">" Value= "6000">
        </ SUBJECT -PROPERTY>
      </CREDENTIAL>
      <CREDENTIAL ID="C6" TYPE= " Doctor>
        < SUBJECT-PPROPERTY ID= "Profession"
OPERATOR= "=" Value= "Doctor">
        </ SUBJECT -PROPERTY>
        <SUBJECT -PPROPERTY ID= "Valid Date"
OPERATOR= "<" Value= "05/10/2000">
        </ SUBJECT-PROPERTY>
      </CREDENTIAL>
      <CREDENTIAL ID="C7" TYPE= "VISA Card" >
        <SUBJECT-PPROPERTY ID= "Expiration
Date" OPERATOR= ">" Value= "01/20/1998">
        </ SUBJECT-PROPERTY>
        < SUBJECT -PPROPERTY ID= "Credit
Value" OPERATOR= ">" Value= "1600">
        </ SUBJECT-PROPERTY>
      </CREDENTIAL>
      </ --Credential set definition-- >
  </ -- Basic Elements -- >
  < !-- Relationships of Elements -- >
    <! --Role hierarchy definition-- >
    <INHERITES FROM = "H" To
"I"></INHERITES>
    <INHERITES FROM = "J" To "I" >
</INHERITES>
    </ --Role hierarchy definition-- >
    <! -- Privilege assignment definition-- >
      <PRIV-ASSIGN ROLE= "I"
PRIVILEGE = "p1 p2"></PRIV-ASSIGN>
      <PRIV-ASSIGN ROLE= "J"
PRIVILEGE = "p3 p4"></PRIV-ASSIGN>
    </ -- Privilege assignment definition-- >
    <! -- Credential assignment definition-- >
      <CONS-ASSIGN ROLE= "H"
CREDENTIALS = "(C5∧C6) ∨ (C6∧C7)"> </CONS-
ASSIGN>
      <CONS-ASSIGN ROLE= "I"
CREDENTIALS = "C1∧C3" > </CONS-ASSIGN>
      <CONS-ASSIGN ROLE= "J"
CREDENTIALS = "C2∧C4" > </CONS-ASSIGN>
    </ -- Credential assignment definition-- >
```

ISBN 0-7803-7808-3/03/$17.00 © 2003 IEEE

< /-- Relationships of Elements -- >
</ORBAC-model>

3.1 Object Model of the XML-based Security policy

For an XML-based ORBAC security policy, such as example 1 security policy, an object model translator is used to automatically create an object model of the security policy which will be used for the following role assignment algorithm. The general procedure of the creation shown in Fig. 3 is described as the following two steps.

Step 1: The object model translator parses each line in part 1 of security policy and creates objects of privileges, roles and credentials based on their class definitions.

Step 2: The object model translator parses each line in part 2 of the security policy, establishes the relationships between role and role, role and privilege, role and credential chains. Finally, an object model of the XML-based ORBAC security policy is created.

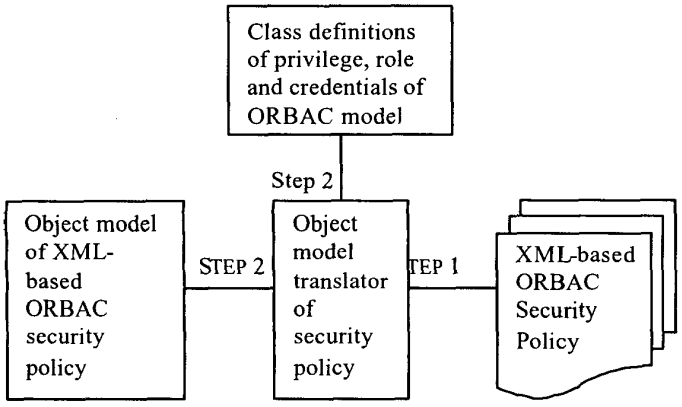

Fig. 3 Diagram of creating an object model of XML-based ORBAC security policy

3.2 Modifying XML-based ORBAC Security Policy

In web-based applications, the number of roles, credentials and privileges can be very large. In general, when some basic elements of ORBAC security policy, such as roles, privileges and credentials are required to be added, modified or deleted, it will be a formidable task. Our method simplifies the maintenance procedure by the proposed XML-based ORBAC security policy. If the security manager adds, deletes or modifies some basic elements or relationships in a security policy, the object model translator will be used to automatically recreate a new object model for the modified security policy.

3.3 Role Assignment Algorithm

Based on the created object model of XML-based ORBAC security policy, a role assignment algorithm is called to authorize roles to Internet users. The algorithm assigns the role that has most privileges to the user based on his/her submitted credentials. We define that a credential in the object model is a *dead-end parent credential* if it has no parent credentials. Also, we define that a credential is a *dead-end child credential* if it has no child credentials. Moreover, we call a credential is a *passed credential* if the credential that Internet user provided satisfies the credential function of the credential in the object model.

In the following role assignment algorithm, three temporary tables (a role table 1, a role table 2 and a credential table) and two stacks (open and closed) are used.

Role Assignment Algorithm

Input: Applied privilege P: the privilege the Internet user applies for.

Credentials: credentials that Internet user provides.

Output: Authorize a role to the Internet user if the submitted credentials satisfy the credential functions of credentials of the role, otherwise, a reject message is returned to him/her.

```
{
    role table 1 []=0, role table 2 []=0;
    For each role S that directly has privilege P
        add role S to the role table 1
    For each role F in the role table 1
    {
        K= Role-Check (F)
        If K≠0
            add role K to the role table 2
        For each direct parent role (if exists) of role F, role G
            add role G to the role table 1
    }
    if role table 2<>0
        Return the role in the role table 2 that has most privileges to the user
    return reject message
}

Role-Check (role L)
{
    credential table []=0;
    open[]=0;
    closed[]=0;
    if there exist credentials for role L {
        Search all credentials of role L that directed linked with role L and add the credential ID of them to the credential table
        For each credential C that credential ID exists in the credential table
        {
```

```
            put the credential ID of credential C to stack
open
            While stack open is not empty {
                Remove leftmost credential ID of credential
X from open
                Implement the credential function of
credential X
                if (credential X is a passed credential and is
a dead-end parent credential)
                    return role L
                    get all parent credentials of X
                    put the credential ID of credential X to
stack closed
            remove credential ID of parent credential of X if
already on open or closed
                    put remaining credential ID of parent
credentials of X on left end of open
                                        }
                        }
                    return 0
                    }
                    return role L
        }
```

For instance, in the object model of example 1 security policy shown in Fig. 4, suppose privilege p2 is applied by an Internet user, the roles that directly or indirectly have privilege p2 are I, J and H, the search path of privilege p2 by the role assignment algorithm is shown in Fig. 4. The search starts from role I that direct linked with privilege p2, if the submitted credentials satisfy the credential functions of credential chain C1 and C3, then role I can be assigned to the user, the algorithm will continue to search its direct parent roles, role H and role J, if the submitted credential satisfy the credential functions of a credential chain C5 and C6 or C6 and C7, then role H can be assigned to the user. After searching all the roles directly or indirectly has the privilege p2, role I, role H and role J, assume role H and role I can be assigned to the Internet user, since role H has more privileges than role I, role H is assigned to the internet user. If the submitted credentials cannot satisfy credential chains of roles I, role J and role H, then a reject message will be returned to the user.

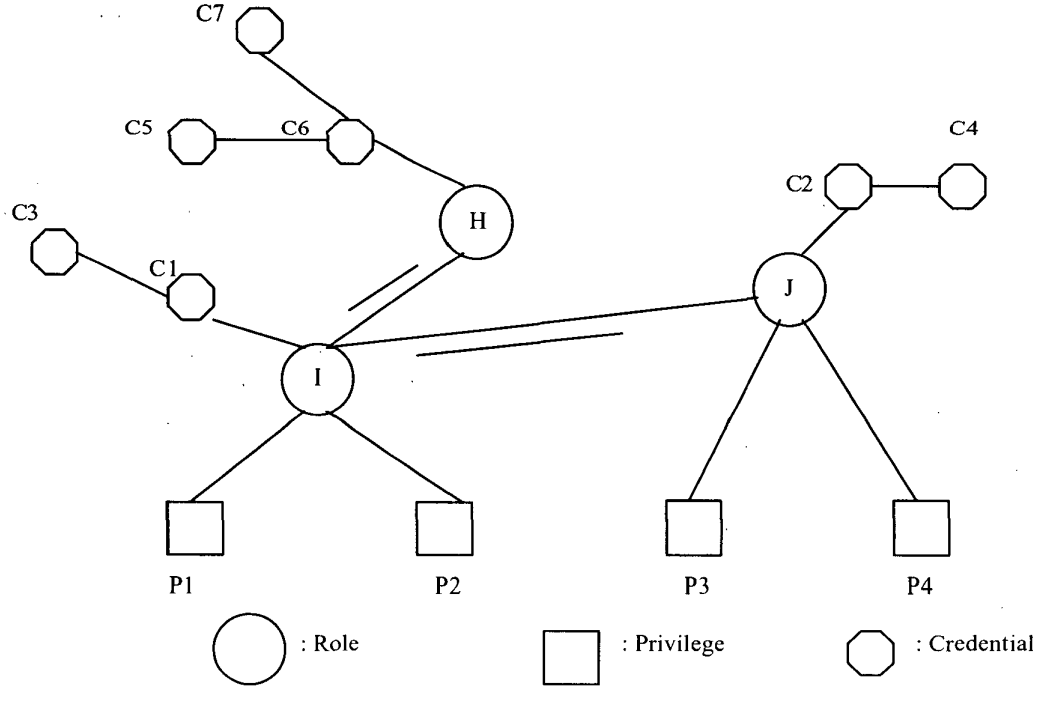

Fig.4. Search path of privilege p2 by the role assignment algorithm

4. Conclusion

In this paper, I have presented the Role-based Access Control model for web-based applications that simplifies security policy administration for web-based applications. Also, using XML-based ORBAC security policy, the authorization is independently defined and is separated from policy representation and from implementation mechanisms. Moreover, a role assignment algorithm is proposed. I believe that the proposed Role-based Access Control model for web-based application and XML-based ORBAC security policy can be applied to express any security policies for web-based applications.

Reference:

[1] R. Sandhu, E. J. Coyne, H.L. Feinstein, and C.E. Youman. Role based Access Control Models. IEEE Computer, 29(2), Febuary, 1996, pp 38-47.

[2] R. Sandhu and Venkata Bhamidipati, The ARBAC97 Model for Role-Based Administration of Roles: Preliminary Description and outline, *Second ACM workshop on Role-Based-Access-Control* , Fairfax, Virginia, USA, November, 1997, pp 41-54.

[3] Ravi sandhu, David Ferraiodo and Richard Kulin, The NIST Model for Role-Based Access Control: Towards a unified Standard, *Fifth ACM Workshop on Role-Based Access Control,* Berlin, Germany, July, 2000, pp 47-64.

[4] Ferraiolo, D.F., Barkley, J.F., and Kuhn, D.R. A role-based access control model and reference implementation within a corporate intranet. *ACM Trans. Info. Syst. Security* 2, 1 (Feb. 1999), pp 34-64.

[5] Nathan N. Vuong, Geoffrey S. Smith, Managing security policies in a distributed environment using eXtensible markup language (XML), Proceedings of the 2001 ACM symposium on Applied computing, 2001, Las Vegas, Nevada, Pp 405 - 411

[6] Extensible Markup Language (XML) 1.0-W3C Recommendation;10-Feb-98.
Http://www.w3.org/TR/1998/REC-xml-19980210.

ISBN 0-7803-7808-3/03/$17.00