Dynamic Role-based Access Control for Web Services using Context and Trust

Abstract

Currently, the security issues of Web services are hot area in information system (IS). This research mainly discusses the key technologies of information access control focusing on following works: After analyzing the dynamic characteristic of application nature for Web services, a Dynamic Role-Based Access Control using Context and Trust model (abbreviated as CT-DRBAC) for Web services is proposed. During Web services, both the subject of invoking request and object of providing service resources are dynamic nature. So, access policies are needed to consider the dynamic nature. The proposed model has been developed and the authorization framework is discussed detail. In order to implement the dynamic trust management mechanism, a dynamic user role authorization algorithm which considers the user lifecycle contexts in the open systems is proposed and designed to meet the dynamic characteristic of subject and object effectively, and achieve intelligent and scientific user role assignments. The proposed access control module can be used in intelligent information systems to grant dynamically roles to users according to the current context.

Keywords: Life Cycle Context, Access Control, Dynamic Authorization Assignment

1. Introduction

Web services are rapidly becoming a fundamental paradigm for the development of complex Web applications. Web services are loosely coupled applications using well-known XML protocols such as SOAP, UDDI and WSDL for representation and communication across the Internet. With the application and development of Web services, resources are shared more widely and efficiently. However, the open nature of the internet and its loop-coupling construction make Web services vulnerable to various types of security attacks. The security of Web services becomes the important factor that restricts Web services further to develop. It is important to develop an effective access control mechanism to prevent Web services from unauthorized accesses and malicious invoking.

Considering user lifecycle contexts, the paper proposes a CT-DRBAC model which is the abbreviation of Dynamic Role-Based Access Control using Context and Trust based on RBAC model for Web services. Comparing with other RBAC models, it is more suitable for the characters of service-oriented architecture for Web services. CT-DRBAC could be used to express dynamic characters of the role to achieve the dynamic authorization by analyzing user lifecycle contexts in the open systems easily.

The rest of this paper is organized as follows: The section 2 briefly discusses the related works and presents the motivation of the paper. The section 3 presents a dynamic role-based access control model. In the section 4, we present a secure architecture model for Web services, and discuss the authorization algorithm. At last, we give a conclusion and point out the problems which should be resolved in further research in the section 5.

2. Related works and motivation

The central notion of RBAC is that permissions are associated with roles, and users are assigned to appropriate roles [1]. This greatly simplifies management of permissions. It is
suitable for Web applications especially, which need to address large-scale user authorization and complex rights management. At present, role-based access control applications in the network access control is becoming the hot topic. Scholars carried out relevant researches on how to apply RBAC in the Web service environments.

At present, main concerns about Web services access control technology are access control for XML documents and SOAP [2,3]. Bhatti, etc., proposed X-RBAC [4], a specification framework supporting XML-based RBAC policy for enforcing access control in dynamic XML-based Web services. But the X-RBAC framework mainly focuses on the XML-based document security, and can not completely realize access control on Web services. In order to solve above issues in the Web services access control, some scholars proposed an extended role-based access control model to meet the needs of distributed network development environments [5,6], which integrate the extensible access control markup language XACML [7,8] with role-based access control model.

With further research, some scholars have found that only focusing on the data storage and transmission file of XML can not meet the security needs, but also need to strengthen the protection of the services. In 2004, Xu, etc., presented a service-oriented role-based access control model and an architecture of secure model for Web services [9]. In the paper, they presented a new technology to implement the RBAC on the Web services by designing the secure cookies and secure SOAP messages.

In 2011, Zhang, etc., analyzed system and access control related context information and proposed a context constraints access control model by using a XML-based policy specification grammar called XGrammar [10]. According to actual demand, Zhang, etc., [11] also established an extended cardinality constraint access control model, and described the user-role, user-session, role-session cardinality constraints based on the RBAC model. M.S.Saleem Basha, etc., developed a web service based secure e-learning management system (EWeMS) which provides an environment for management and e-learning strategy [12]. In this system, they proposed a concept called Expected Clandestine Figure (ECF) for default authentication of the service consumer in a closed group environment.

Moreover there are still many issues needed to be addressed. Above the researches, only take account into the user access control levels from the message, but not consider the user's dynamic characters. In order to address the existed problems, we propose a Dynamic Role-Based Access Control using Context and Trust model (CT-DRBAC) for Web services.

3. CT-DRBAC model

Web services can support dynamic service-oriented architecture, but traditional RBAC can not express dynamic execution process to access services. So we propose a Dynamic Role-Based Access Control Using Context and Trust model (CT-DRBAC) for Web services, shown as Figure 1.
The authorization can be described by the following entity sets, CT-DRBAC = \{Rq, Cxt, T, Rr, P, C, S, AD\}. The Formal description of each element is as follows:

- **Rq** -- Requestors: It is the set of all requestors of Web services who will access some operations and services provided by providers of Web services. \( Rq = \{rq_1, …, rq_n\} \).

- **Cxt** -- Contexts: A collection of properties and their values, which restricts the user role assignment and decision making. \( Cxt \) is the set of contexts.

- **T** -- TrustValue: In CT-DRBAC model, there is not exist the directly assigned relationship \( T \) between requestors and roles, but a Web services requestor is assigned to a role or a set of roles according to its trust value. The range of trust value is a real number between 0 and 1. The function \( T(rq) \) returns the trust value for the current requestor \( rq \). Where \( T(rq) = \{t \in \text{TrustValue} | rq \in Rq\} \) is the expression.

- **Rr** --Req_Roles: It is the set of all roles assigned to requestors of Web services. The role reflects the relationship that Web services provider has with Web services requestor. \( Rr \) is the set of roles. \( Rr = \{rr_1, …, rr_n\} \).

- **P** --Permissions: It describes all operations applied to objects. \( P \) is the set of permissions.

- **C** --Constraints: \( C \) is a series of constraints to control the assignment of operation, which specifies the separation of duties and avoids conflicts. In the model, the constraint mechanism takes on contexts.

- **S** --Services: A service is an abstraction of the operations provided by the system with its resources. Formally, a service is a subset of the operation set \( P \), and is designated by the service name \( S = \{s_1, ..., s_k\} \), where \( s_i \) is a service, and \( i = 1..k \).

- **AD** --Access Decision: It is the set of all access decisions that can be made by services provider in the Web system.

In the CT-DRBAC Model, assigned relations include \( UA \) (User Assignment), \( PA \) (Permission Assignment), \( RS \) (Role-Service Relation), and \( RH \) (Role Hierarchy). The specific description is as follows:

- **UA**: \( UA \subseteq U \times R \), assignment relation from user to role with many-to-many mapping.

- **PA**: \( PA \subseteq P \times R \), mapping relation from permission to role with many-to-many.

- **RS**: \( RS \subseteq R \times S \), a many-to-many role service relationship.

- **RH**: Role hierarchy is a strict partial order relationship. The superior role can own the permissions of subordinate roles.

### 4. Service-oriented role-based access control

#### 4.1. Context access control

In order to formalize the context, we introduce the parameter information to allow specifying domains of legal values for various context parameters. Our formal model relies on the components and can be defined as follows:

- **Parameter Name Set**: A set \( PN \) denotes the possible names of context parameters.

- **Parameter Type Set**: A set \( PT \) denotes the possible types of context parameters.

- **Context Parameter**: A context parameter is represented by a data structure \( cp \), which has following fields: \( \text{name} \in PN \), \( \text{type} \in PT \), and a function \( getValue() \).

**Definition 1 (Context set)**. A context set \( Cxt \) consists of \( n \) context parameters \( \{cp_1, ..., cp_n\} \), \( n \geq 0 \). For any \( cp_i, cp_j \) with \( i \neq j \) and \( 1 \leq i, j \leq n \), exist \( cp_i.name \neq cp_j.name \) (i.e. the parameter names must be distinct).

\( PN \) and \( PT \) we mention here constitute a set of pre-specified parameter names and types determined by the System Security Officer. The \( cp.getValue() \) function can be used to dynamically compute the value of the parameter, and its invoking is done by independent system.

Currently, the definitions of context given by Abowd and Dey [13,14] have been generally recognized. They thought that "the context is any information which can be used to characterize the situation of entities (i.e., whether a person, place, or object) and has considered relevant to the
interaction between user and application. It includes the user and the application themselves. Context is typically the location, identity, state of people and groups, and computational and physical objects”.

On the basis of the definition proposed by Abowd and Dey, we can classify the context into two types of user context and service context. Service context and user context are correlative dependence, and the user contexts depend on the service contexts. The partition of context can maximize the use of access control information, which provides a more detailed assessment to the user role assignment. So it could be used to improve the accuracy of the user role assignment.

4.2. Trust Evaluation Model

The traditional trust and reputation mechanisms lack of flexibility in modeling of diversity and dynamicity of trust in Web services environment. In this section, we try to evaluate trust with the context and adopt a context feedback control mechanism to realize the assessment of trust in which a set of subjective weights are set and adjusted in time to reflect the dynamicity of trust environment. The trust management for requestors is involved with trust context.

The architecture of feedback control mechanism is shown as Figure 2. The loop of trust evolution is emphasized within a link of Inputs, Trust Compute Unit, Decision Executing Point, Context Feedback Point, and Trust Policy.

![Figure 2. Trust Evaluation Model](image)

The trust management schema is a dynamic procedure for adapting trust value with a set of weights. The Context Feedback Point is in charge of adjusting the weights according to the set of contexts and trust policy. Decision Executing Point judges the role of the requestor through the maturation of trust. The trust value is the sum of the former value multiply by different weights. Finally, whether a trust behavior is triggered is decided by the Decision Executing Point.

A formalized expression of trust computation in trust computation unit will be discussed detailed in following.

**Definition 2 (General Trust Value).** The formula of general trust value is defined as follows:

\[
T_u (< t_1, c_1 >,< t_2, c_2 >,.....,< t_m, c_m >) = \sum_{i=1}^{m} w_i c_i \text{-index}(i) .
\]

(1)
In above formula, $<t_m, c_m>$ is a two-dimensional array, where $c_m$ is the user context and $t_m$ is the timestamp of context. The variable $t\text{-index}(i)$ is the index of the object with the $i$-th largest of the timestamp $t_i$. $W^* = (w_0, ..., w_m)^T$ is the transposition of context weight vector ($0 \leq w_i \leq 1$, $\sum w_i = 1$).

4.3. Access Control Decision

Access control rules were represented as logical expressions in [15, 16]. How to make an access control decision, a query? $\text{dercando}(\text{wsobject}, \text{wsrequestor}, +\text{saction})$ is answered by a process of inference. For example, the query $\text{dercando}($SilentTribute, $r_q$, $+\text{exe}$) determines whether Web services requestor $r_q$ can execute the silenttribute operation. Access control rules and facts will be described in next.

Assume that moderately trusted Web service requestors may be granted access to the SilentTribute, CancelOrder, Mourning operations. SendRose is an operation granted only to highly trusted role for Web services requestors. Assume there is a requestor named $r_q$, who is moderately trusted role. The roles are defined in the systems as follows: the associate role has a moderate trust level (moderate = 2) and the partner role has a high trust level (high = 3). Next, we will illustrate the operation of the access control policies, which is shown in Figure. 3.

In the policies, a Web services requestor may access an operation, if the Web services requestor is active in the role to which the permission has been assigned shown by rule 1. Rule 2 determines the role of requestor by comparing the trust levels of the Web service requestor and the trust scope of role defined by system. Trust levels are retrieved from the trust component when facts are instantiated. The predicate can be used to allocate the role $r_r$ to access the object. For example, assume $r_q$ needs to execute the operation SilentTribute. Facts are instantiated in the access control policy:

$$cando($SilentTribute$, $associate$, $+\text{exe}$).$$
$$reqtl(r_q, 2).$$
$$roletl(associate, 2).$$

This section provides an expanded system architecture which illustrates the interplay between the features of context and trust adjustment discussed above. It also provides general design guidelines for implementing the trust-based context access control model in real Web services environments. The system architecture is depicted as Figure 4.

The Figure 4 shows the RBAC processor how to carry on identity certification and role judgment. The detail of the procedure is as follows:

1. **AEP** (Access Enforcement Point) is the entry point of all the requesting processing, which is between the input and output function module in the access control module. First, **AEP** judges whether the user context base has been updated according to the service context information. Then, it will determine the specific safe handling procedures by the service context information. If the user context has been updated, it will send the user requesting information to access.
control policy enforcement module to assign role for the requester by the related assessment strategies; Finally, AEP compares the role permissions with the user's request. If the permissions allow users to access the requested service, it will permit the request. Otherwise, it will reject the request.

2) Access control policy enforcement module is responsible for authorization management, context information collection, the implementation of access strategy. It receives access requisition coming from AEP, and uses the trust evaluation algorithm according to the context of the access request to evaluate the trust value, and implement dynamic user role assignment. The following is the implementation of access control policy of each sub-module functional components specified.

![Figure 4. An Architecture for Web Services Access Control](image)

(3) PPC (Prepare-Contexts) is responsible for analyzing service context information from the AEP, and the results are returned to the context of pre-processor CxtProc. Pre-processor CxtProc gains user context according to serving context.

Access decision point is the core of access control policy enforcement module, which used for the trust evaluation and user authorization management

URA (User Role Assignment) queries the corresponding role from the date base of role-trustiness by the trust value, and then sends them to AEP which will control the invocation of Web services according to the role of requestor.

In practice, the user contexts may be maintained as a cache, so that the consistency analysis may be carried out offline. The resource accesses may be prioritized so that only the more sensitive ones can analyzed in real-time, and the rest can be handled offline. This would make the mechanism more feasible in real systems.

5. Conclusion

The paper presents a Dynamic Role-Based Access Control using Context and Trust model (CT-DRBAC) for Web services through consider user lifecycle contexts, and a Web services access control secure architecture as well as the authorization algorithm in detail. Because CT-DRBAC model is
extended with the description for dynamic characters of Web service, so it can be used to realize the dynamic authorization by analyzing the relationship among roles based on user contexts in s/he lifecycle. It is more suitable to the Web service applications with the features of dynamic and large numbers of users. However, the paper does not address the authorization management of different autonomous systems. This is one of our works in future.

6. Acknowledgement

This paper was supported in part by the National Natural Science Foundation of China (No. 60874120), and Innovation Program of Shanghai Municipal Education Commission (No.11zz188).

7. References