I.J. Engineering and Manufacturing **2011, 2, 54-59**
Published Online April 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijem.2011.02.09

Available online at http://www.mecs-press.net/ijem

A Measurable Approach for Access and Control Modeling in SOA

Gu JunKai[a], Han Ling [a], Wang Bo[a]

[a] *College of Geology Engineering and Geomatics, Chang'an University, Xi'an, China*

Abstract

Service oriented architects (SOA) is considered as an extensible, robust and platform independent web application architect. However, problems in security guard for service access remains unresolved especially for the measurable one. we proposed a novel access control model, which we called SACM: Service Access Control Model, specially for SOA. Our model is mainly based on the role access policy, extended with trust authority transition and integration mechanism, to fulfill an extensive and measurable access control modeling approach with Crypto-CCS.

Index Terms: Component; SOA; Access control;Trust

1. BackGround

A service oriented architect (SOA) is considered as an extensible, robust and plate form independent web application architect. In which final applications are constructed by services that located on internet queried by UDDI server to meet with the customer needs. With its open and distributed properties, services and resources, such as common users, facilities, database, services etc that categorized into services providers, service requesters, and service entities, on internet need a security mechanism to avoid illegal accessing. Security properties that related above mainly referred to integrity and confidentiality. Integrity means information flow will not be destroyed by illegal or unauthorized entities. Confidentiality means protected information will not leaked or accessed by illegal or unauthorized entities. In order to guarantee those Security properties, researchers have contributed lots of works. DACM: Discretionary Access Control Model [1] was proposed by trusted computing group of US national defense that allow authorized users or user group to access corresponding resources. The Discretionary Access Control process can be simply depict as two step, (1)Verifying identity of demander from existed authentication database.(2)according to identity information, query related authority from authentication database. DACM, on the other side, allow certain users to grant authority of accessing certain objects to other users independently. MACM: Mandatory Access Control Model,

* Corresponding author.
E-mail address:

another resources accessing policy, which also proposed by trusted computing group of US national defense, provided a more strict rules to protected classified military documents from illegally accessing, and then widely used in open industry. In MACM model, the accessing demands that from any subject should obey two rules,(1)security level and(2)security class. Security level is a group of binary lattice relation which describes the accessing authority that subjects accessing object. While security class describes the security category that each subject and object belongs to. There are four different accessing tactics to obey when a subject is trying to access an object: up and down ward read, up and down ward write. How ever, DACM and MACM model can only satisfy integrity or confidentiality, take an example, Biba Model [2] is a typical MACM model in which prevent object from reading down ward and writing up ward, thus can only guarantee integrity. Bell-LaPadula Model [3] allows reading down ward and writing up ward, thus only guarantees confidentiality. Role Based Access Control model [4, 5], was proposed by SandHu who abstracted users by related transitions or functions as roles. Authorities are bounded with roles and separated with single users in order to decrease the complex of authority management. In SOA framework, resources and services that distributed on internet are visited ambiguously every time, and may cross different security region, thus need a more flex, measurable and complete solution. DACM or MACM model are too strict and not complete for SOA authority management, while RBACM model lacks of flexible and measurable mechanism. In this paper we proposed a novel access control model, which we called SACM: Service Access Control Model, especially for SOA. Our model is mainly based on the role access policy, extended with trust authority transition and integration mechanism, to fulfill an extensive and measurable access control modeling approach. As in SOA framework, an application is composted by single services, and services may communication with each by message, thus we shall use Crypto-CCS [6, 7], a special process algebra as our basic modeling language.

2. Service Access Control Model

In this chapter we show details of our model. Our model is separate into several parts. First of all is simple access condition: a subject role of a service can not request a certain service to access certain resources (objects) roles within a certain security tag range.

$$s_i \underrightarrow{oo}_j \Rightarrow il(s_i) \underrightarrow{leoil}(o_j) \tag{1}$$

The above formula demonstrates the simple access condition rule of SACM, in which s_i means subject role, o_i means object role that denote some kind of resource to be accessed. Function $il(r)$ depict the security tag of both subject and object role from an existed database and if s_i could access o_i implements that $il(s_i)$ and $il(o_j)$ satisfy binary relation $\underrightarrow{leo}$.

Access property: a subject role of a service can not request a certain service to modify certain resources (objects) roles within a certain security tag range.

$$s_i \underrightarrow{mo}_j \Rightarrow il(s_i) \underrightarrow{lemil}(o_j) \tag{2}$$

The above formula demonstrates the access property rule of SACM, in which subject s_i could modify object o_i implements that $il(s_i)$ and $il(o_j)$ satisfy binary relation $\underrightarrow{lem}$.

Invocation property: a subject role of a service can not demand a secondary subject to invoke a certain service.

$$s_i \underrightarrow{is}_j \Rightarrow il(s_i) \underrightarrow{leiil}(s_j) \tag{3}$$

The above formula demonstrates the Invocation property: rule of SACM, in which subject s_i could demand another subject s_i to invoke a certain service implements that $il(s_i)$ and $il(s_j)$ satisfy binary relation $\underrightarrow{lei}$.

The SOA final application is composted by single services, if it was verified that every service with its subject and object satisfy those rules, then resources accessing is legal. However, those above are under the assumption that authority information is constant. Thus in SOA application, resource access or service demand would cross security region, which means authority should be transmitted or modified in such certain conditions. Thus a more flexible and measurable mechanism is needed. For this purpose, we add trust scale with its operation to our model above. First of all, a trust value which indicates a confidence that a subject role grant another role to resolve a task. Trust value can be modified or calculated dynamically due to different situations. We list the rules for trust value calculates as below:

(1) $s_i \xrightarrow{f,v} s_j$: simple trust rules

(2) $s_i \xrightarrow{rf,v} s_j$: simple trust recommendation rules

(3) $$\frac{s_i \xrightarrow{rf,v_1} s_j \quad s_j \xrightarrow{rf,v_2} s_k}{s_i \xrightarrow{rf,v_1 \otimes v_2} s_k}$$: trust transition with recommendation rules

(4) $$\frac{s_i \xrightarrow{rf,v_1} s_j \quad s_j \xrightarrow{f,v_2} s_k}{s_i \xrightarrow{f,v_1 \otimes v_2} s_k}$$: common trust transition rules

$$\frac{s_i \xrightarrow{f,v_1} s_j \quad s_i \xrightarrow{f,v_2} s_j}{s_i \xrightarrow{f,v_1 \oplus v_2} s_j}$$: trust integration rules

Rule (1) indicates that a subject role s_i would trust another subject s_j in trust value v to invoke some service in order to complete task f.

Rule (2) indicates that a subject role s_i would trust another subject s_j in trust value v to recommend some thirty parties r to invoke some service in order to complete task f.

Rule (3) indicates a transition policy that a subject role s_i would trust another subject s_j in trust value v_1 to recommend some thirty party r to invoke some service in order to complete task f, while subject role s_j would trust another subject s_k in trust value v_2 to recommend the same thirty party r to invoke some service in order to complete task f, then have a conclusion that subject role s_i would trust another subject s_k in trust value $v_1 \otimes v_2$ to recommend some thirty party r to invoke some service in order to complete task f. Symbol $\otimes$ is a binary transition operator for trust value that would be decided practically.

Rule (4) indicates a integration policy that a subject role s_i would trust another subject s_j in trust value v_1 to invoke some service in order to complete task f, while subject role s_j also trust subject s_j in trust value v_2 to invoke some service in order to complete task f in another environment (this would concern

with security region crossing), then have a conclusion that subject role s_i would trust subject s_j in trust value $v_1 \oplus v_2$ to invoke some service in order to complete task f. Symbol $\oplus$ is a binary integration operator for trust value that would be decided practically.

3. Embed SACM in service composition model

The SACM model demonstrated above offers a flexible mechanism for resources accessing and authority decision, especially for service demanding that crossing security region in SOA framework. However, The SOA final application is composted by single services, thus we integrate our model with a service composition model in order to gain a final solution. We translate our model into Crypto-CCS a special process algebra that provides a deduction mechanism for our modeling language. The SACM model described in Crypto-CCS deduction rules are showed below:

$sic: \dfrac{\{m, m'\}, m\ leo\ m'}{0}$: Service Access Control Model

$intp: \dfrac{\{m, m'\}, m\ lem\ m'}{0}$: Access property

$invp: \dfrac{\{m, m'\}, m\ lei\ m'}{0}$: Invocation property

$ttr: \dfrac{A \xrightarrow{r\,f, v_1} BB \xrightarrow{f, v_2} D}{A \xrightarrow{f, v_1 \otimes v_2} D}$: Common Trust Transition Rules

$rtr: \dfrac{A \xrightarrow{r\,f, v_1} BB \xrightarrow{r\,f, v_2} D}{A \xrightarrow{r\,f, v_1 \otimes v_2} D}$: Trust Transition with Recommendation Rules

$tir: \dfrac{A \xrightarrow{f, v_1} BA \xrightarrow{f, v_2} B}{A \xrightarrow{f, v_1 \oplus v_2} B}$: Trust Integration Rules

In Crypto-CCS, a single service is modeled as an action, and SOA application P is considered as a transition system $(S, A, \rightarrow, S_0, nil)$ in which S is a set of stats, and A is a set of actions, and P can also written as $P = \sum a_i.P_i$ in which single service a_i can either be $c(x)$ or $\bar{c}m$, which means messages can be transmitted on channel c. SACM rules are applied in the form of $R = [< m_1, ..., m_r > |-_{rule_a} x]P; Q$, which means if context $< m_1, ..., m_r >$ of P satisfy SACM $rule_a$, then service state P is successful evolved into Q, otherwise, P would be terminated.

4. Using in practice

In this chapter, we discuss about an example that we have applied our model in He Nan province Jiao zuo Government project approval system. This system was constructed in SOA framework, with multiple users and project documents in complex documents accessing authority management, thus need a robust accessing control module. In this system, SAMC rules are coded in XACML (Extensible Access Control Markup Language) [8], which is stored in central server. Context information such as user confidential and trust value

in certain security region is bonded and transmitted in SOAP messages. The total design of accessing control module is showed in Fig. 1. As space limits, more details would be discussed in our sequent papers.

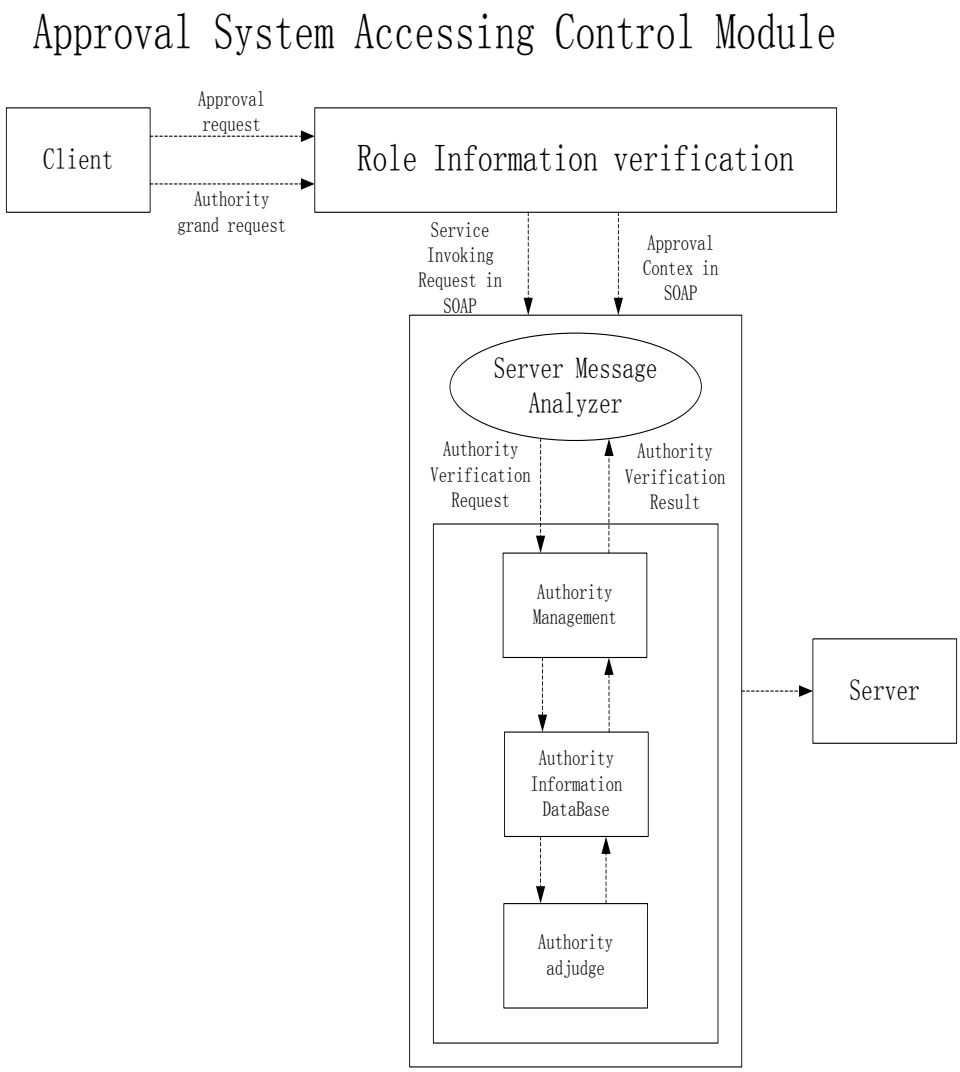

Fig 1 Total Design of Accessing Control Module

5. Conclusion and further works

We proposed an accessing control model for SOA framework. Our model is mainly based on the role access policy, extended with trust authority transition and integration mechanism, and embedded in Crypto-CCS. Compared with other works, our model is more flexible and trust in service and resource accessing can be measured. We also applied our model in practice and achieved a successful feedback.

More over, we did not concern about accessing rule conflicts and evolvement, this may be contained in our further works. On the other side, as XACML is an open and extensible language, thus more rules can be added in our model and use in practice, perfect our model in more situations is another direct of work.

References

[1] James B. D. Joshi,Walid G. Aref,Arif Ghafoor, Eugene H. Spafford. Security models for web-based applications.Communications of the ACM. ACM New York, NY, USA.44(2).38-44.2001
[2] Ravi S. Sandhu.Lattice-Based Access Control Models. Computer. 26(11).9-19. 1993
[3] D Bell.The bell-lapadula model.Journal of computer security, 1996
[4] Sandhu R, Coyne E J, Feinstein H L, et al. Role-Based access control models[J]. IEEE Computer, 1996, 29(2): 38-47.
[5] Youman C, Sandhu R, Coyne E, et al. Rationale for the RBAC96 family of access control models[A]. In Proc. Of the 1st ACM Workshop on Role-Based Access Control[C]. New York: ACM Press, 1996.
[6] D. Marchignoli and F. Martinelli. Automatic verification of cryptographic protocols through compositional analysis techniques[J]. In TACAS, volume LNCS 1579, pages 148–162. Springer, 1999.
[7] F. Martinelli. Analysis of security protocols as open systems[J]. Theoretical Computer Science, 290(1): 1057–1106, 2003.
[8] Markus Lorch, Seth Proctor, Rebekah Lepro et al. First experiences using XACML for access control in distributed systems.in: Sushil Jajodia, Michiharu Kudo eds[C]. Proceedings of the 2003 ACM Workshop on XML Security. New York, NY, USA: ACM Press, 2003: 25–37.