A Policy Driven Authorization Control Framework for Business Collaboration

Daisy Daiqin He
Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
{daiqin, jian}@ics.mq.edu.au

Jian Yang

Abstract

Business collaboration is about coordinating the flow of information among organizations and linking their business processes. However unique and challenging security issues concerning business collaboration in the context of SOC have arisen because of its dynamic and loosely coupling nature. In this paper we firstly identify elements for security policy specification. We then discuss different types of collaboration patterns based on which rules for security policy integration can be developed. A framework is developed for handling authorization control for business collaboration. The discussion in the paper is based on the collaboration scenarios in health care.

1. Introduction

Web services and Service Oriented Computing (SOC) provides infrastructural support for cross-organization collaboration in distributed environments. Each organization or business unit has its own interest and security polices for defining who has access right for specific services and how services can be used. In web services environment with complex cross-organization collaborations, how security policies can be propagated or integrated is a real challenge. We shall use some examples in health care to illustrate some issues.

Patient needs to disclose his/her health records to a GP for diagnosis purpose. But in order to protect his/her own privacy, before the patient disclose records to GP, the patient could restrict the way that the GP forwards his/her health record to a third party.

Policy inconsistency among participants is another problem area in the collaborative world. For example, a patient who holds an OSHC (overseas students health cover) wants to book an appointment with a physician online for general enquiry and ultrasound examination. The physician needs to collaborate with a radiology institute for this patient. OSHC is acceptable for the physician; however it is not an acceptable health cover for the radiology institute. If both parties want to collaborate, they must negotiate to reach an agreement.

Most security control and policies are discussed and specified from the perspective of service provider. However business partners collaborate in a peer-to-peer fashion, which means security policies of both provider and requester need to be considered. For example, staff of medical insurance company wants to access patient records for validation purpose, medical center asks ID of this staff as credential, but disclose ID is not permitted in the security policy of insurance company. In this case, security policies of both parties need to be considered. The issue will be further complicated if a requester comes from a cross-organization collaboration environment.

To address these complex security policy issues, we need a framework that can analyze, validate and integrate security policies if necessary for collaboration purpose, based on which negotiation can be guided and security integrity can be enforced.

Research has been done on the low level security issues in terms of protocols or security specification languages. In [2], author focus on policy language that implemented an authorization service on Microsoft .NET MyServices. Most of studies focus on message level security or how to specify security information in message by extending existing language, which is not our focus in this paper.

A few of studies have touched policy level security issues, which focused on identifying different
security requirements and proposed specifications for these requirement. Trust-Serv [3] modeled access control processes in web services using state machine and provides lifecycle management for policies. Ws-AC [4] provides an adaptive system that is capable of asking users to refine their requests to comply with security policies. All of them are concerned security policies in a single organization and none of them addressed policy problems in collaborative environment.

Furthermore, these studies only focused on providing solutions to some aspects of security issues in terms of: security policy specification, access control in distributed environment, and access decision making. What is missing is a comprehensive analysis of: (1) what security requirements really are in the context of business collaboration; (2) security policies can be specified; (3) how security policy can be verified, compared, and incorporated/integrated for the purpose of business collaboration. No feasible mechanisms can be developed for policy negotiation and enforcement without this analysis. And in this paper we present some of our initial results in modeling security requirements and integrating security policies for business collaboration. We believe this study is the first step towards achieving an understanding of a secured of business collaboration in terms of authorization control.

We first explain the basic elements included in the security policy in the following section.

2. Basic Concepts and Elements in Security Agreement

As we mentioned above, each organization has its own security policies. We assume that before collaboration starts, certain forms of agreement must be established among all the participants regarding each other’s security policies. We call it security policy agreement, which can also be a process of integrating all participants’ security policies.

For this purpose, we define the elements that need to be specified in an agreement: roles, services and organization generic security rules.

Roles
A role is a job function within the context of an organization with some associated semantics regarding to the authority and responsibility conferred on the user assigned to the role. [1] (E.g. “doctor” is a role, John is an actual user, and John is a “doctor”).

Privileges
We use privileges and permissions interchangeably in this paper. Privilege is an approval to perform an operation on one or more objects, such as: read, write, forward, delete etc. In role based access control, permissions are assigned to roles rather than individuals [1], for example, security policy will specify that “Doctor” has “update” privilege on patient health records instead of “John” has “update” privilege on patient health records.

Credentials
Credentials are signed assertions describing attributes of the owner [3]. Examples of credentials are digital certificate, registration number etc. Credentials are used to gain access right. In our framework, a user need disclose certain credentials in order to be able to play a “role” and in turn to get privileges on certain services that assigned to the role.

Services
Services are self-contained processes [6]. Services are accessible objects for users, which could have provision or obligations [7] (we also call them conditions) associate with it. Before certain privilege is granted to requester, certain conditions need to be applied and agreed upon, e.g. agree not to forward to third party. Whether a condition needs to be agreed before access is granted depends on each individual services or resources. The relationship among roles, conditions, credentials and services are illustrated in Figure 1.

In the proposed framework, there are three types of policies defined as follows:

\[
\text{Policy} \rightarrow \text{RP} \mid \text{SP} \mid \text{OP} \\
\text{RP} \rightarrow \text{Privileges} + \text{Credentials} \\
\text{SP} \rightarrow \text{Privileges} + \text{Conditions} \\
\text{OP} \rightarrow \text{Organizational general security policy}
\]
• Role security policy (RP), which specify privileges and required credentials of each role. E.g. Security policy of role “Doctor” states that doctor has add, update, forward, review privileges on patient record, required credential for role “Doctor” is Doctor ID;
• Service security policy (SP), which specifies provisions or obligations associated to each privilege of each service. E.g. Security policy for “Patient Health Record” states that whoever requests “forward” privilege on this service must agree to protect patient privacy;
• Organization general security policy (OP): since organizations can have generic security rules applied to all circumstance, e.g., X.509 is the only accepted certificate format in the organization, which means other organizations must get X.509 certificate in order to interact with this organization.

3. Categorization of Collaboration Types

In order to understand the security requirements for business collaboration, we need to analyze the way organizations collaborate with each others. In Figure 2, we identify three types of collaboration between organizations in the context of SOC.

Type A depicts the most basic scenario in which two organizations interact with each other directly. In figure 2, organization A wants to access services provided by organization B. E.g. health cover company wants to access patient treatment record from a physiotherapy clinic for claim validation purpose.

Collaboration Type B is a scenario of service propagation: organization C granted the access right to a service to organization B, and organization B forwards this access right to organization A, e.g. Patient grant physician access right to his/her health record, physician forward this access right to staff in emergency room when there is an emergency.

Types C1 and C2 in Figure 2 are two forms of service composition. Service composition we discuss here is the situation when organization A wants to combine its services with services provided by organization B, and provides this combined service to third parties. C1 and C2 depict two possibilities existing in service composition:

C1: Service collaboration with an agent
Organization C works as an agent accessing services from different providers, e.g. organization A and organization B, and combine these services together to provide a composite service to other parties.

C2. Joined collaboration without an agent:
Organizations A and B provide a composite service by combine their services or part of their services together directly without involvement of an agent. We can find this practice in most medical centers.

In the following section, we model these different types of collaboration in the domain of Health Care.

4. Security Control Modeling in Health Care

4.1. Basic Access Model

This is a quite straightforward case. When Organization request a service from organization B, the security policies need to be met is organization B’s general security rule $\cup$ required credential for specific role $\cup$ agree with provisions and obligations for required services (if there are any).

Figure 3 depicts security policy interactions between two Health Care organizations. Health Cover company is a requester while service provider is a medical centre. We assume medical centre has a general security rule: X.509 is the only acceptable certificate. All the requests have to satisfy this general rule. Two roles are identified in medical centre: visitor and patient authorized visitor. Anyone who can provide a valid X.509 certificate could become a visitor. A visitor can perform simple browse and search actions. The role as a patient authorized visitor has privilege to access patient health records. However to become a patient authorized visitor, a visitor must
submit a “patient consent”, which is a form of patient agreement that shows patient agreed to disclose his/her records to the visitor/ requester. In addition, there is a provision required for access patient health record, i.e., anyone who wants to access this service must first agree to protect patient privacy. Another party in this example is a health cover company. Two roles identified in this health cover company; Claim Dept staff and department manager of Claim Dept. Assumed security rules here is department staff needs provide a special password to become a department manager in the system, and only Claim Department manager has the privilege to initiate a request to access patient health record for validation purpose. Health Cover Company does not have specific general security rules. In this case, in order to access patient health record, the claim manager first needs to obtain a valid X.509 certificate and must possess a form of consent from patient to become the patient authorized visitor. Lastly, Claim Dept manager in Health Cover Company needs to agree with the provision set for patient health record before he can actually access the service.

![Figure 3. Basic Security policy interactions](image)

### 4.2. Service Propagation Model

Figure 4 models interactions among three or more parties in the service propagation situation. For example, a Patient discloses or provides his/her health information to the General Practitioner in a Medical centre together with his consent (security policy agreement) stating that only GP is allowed to update the records and disclosing to any third parties is not allowed unless there is an emergency. Based on this agreement between the Medical centre and the patient, the patient’s health record could be forwarded to hospital emergency room when there is emergency requiring relevant information. “Emergency” is the compulsory requirement here. In addition to the “emergency” condition, the hospital also needs to present a X.509 certificate for authentication purpose. Moreover, provision about protecting the patient privacy still need be agreed on before the health information to be forwarded to emergency room in the hospital. In another case, the pathologist in a pathology institute wants to access patient records for certain reason, he/she sends a request to the medical centre, but the request is denied although pathologist has a valid X.509 certificate and agree with the provision of protect patient privacy since this request does not meet the disclose condition specified by the patient.

![Figure 4. Policy integrations in service propagation](image)

### 4.3. Agent Based Service Composition Model

Referring to above example, in service propagation situation, assumes A is the original provider, B is the first level requester, and C is the second level requester. Security policy integration rule that organization C has to satisfy is: A’s Policy ∩ B’s policy; A’s provision ∪ B’s provision if there is any.

Figure 5 depicts a scenario that will possibly happen in near future as the development of information system in health care industry. Health care staff can access various information system services (e.g. EAP – Electronically Assisted Prescription system, computer aided diagnosis services etc.) through a medical information system portal. Services available in the portal are from different health care units. Each unit containing medical resources is autonomous and has controls over its data. Each service providers and the portal have their own security policies regarding different services. We assume X.509 is the acceptable certificate format in all participating parties as a generic security rule. Security policy in Health care unit A and B for access EAP or Computer diagnosis system is: the Physician with valid physician registration number has privileges to use the system. Based on security policy of first level service providers, medical IS portal adds this policy to its policy store for accessing these services. Medical IS portal may have its own provision on these services: requester must agree to protect patient privacy before accessing any information from the portal. Therefore, in order to access EAP or CADS, a requester need to present his X.509 certificate, valid physician registration number has privileges to use the system.
registration number and agrees to protect the patient privacy.

When services are composed with an agent, the integrated security policy of the agent should be the base service provider’s policy $\cup$ agent’ policy; Agent’s provision $\cup$ first level provider’s provision if there is any.

### 4.4 Joined Service Model

It is common that a specialized clinic needs to cooperate with a pathology institute, specialist and pathologist to provide joined services to patients. In figure 6, we assume both specialized clinic and pathology institute have the same generic security rule: ask for X.509 certificate. In this case X.509 is not held by any party, it needs to be obtained from an authority. According to security policy of the specialized clinic: specialist has privilege to access all the patient health records as long as the specialist agreed with pre-defined provisions. The security policy of pathology institute is: only pathologists and referral doctors have privileges to access patient pathological records if the requester agreed with provisions on pathological records. In this case, specialist who deals with the patient from Specialized clinic also plays a referral doctor’s role in pathology institute for the patient; therefore the specialist also has privileges to access pathological records. However according to the security policies of both parties, pathologist does not have privileges to access all the patients’ health records in a specialized clinic. The security policies of pathology institute and specialized clinic are not completely compatible according to above example. Security policies for joined service can only contain policies acceptable for both parties since policies that are not acceptable for any participants should not become policies for joined service.

Therefore, security policy integration rule for joined services should be: A’s security policy $\cap$ B’s security policy; A’s provision $\cup$ B’s provision for specific services if there are any.

### 5. The PD-AC Framework

The core function of our PD-AC (Policy driven authorization control) framework is realized by a Policy Controller, which is associated with every organization that provides services in business collaboration, see Figure 7. Policy Controller is used to analyze the nature of collaboration and determine the integrated security policies. Once receiving a request for a service, the Policy Controller firstly identifies the type of requested service, which is a process that identify whether the requested service is a collaborative service. If it is a simple service that does not involve any collaboration, the controller will perform normal comparison: looking up security policy database, check which role in the database has the requested privilege; compare requester’s credential with security requirement; make access decision.

If it is a collaborated service, the controller will perform the following tasks:

- Identify collaboration type;
- generate integrated security policies that the integration rules as discussed in the previous section;
- Compare requester credential with integrated security policies requirements;
- Make access decision.

Based on this framework and elements we used in the framework, we shall provide the formal notations for the concepts and rules discussed before in the following section.
6. Formalization of Policies and Security Control

As discussed in the previous section, security policies can be classified into three groups: role security policy, service policy and organization general security policy. From business and database management point of view, this classification makes it easy for comparison and mapping between elements. It also illustrates the complex relationships among these concepts. In this section we shall give formalization for our proposed framework.

6.1. Formalization of Policies and Access Request

**Definition 1 Role Security Policy**

This policy is used to clarify the requirements to play a certain role and privileges/access permissions assigned to the role. When we receive an access request we first identify which roles have desired privileges to requested service using these roles policies; once we find the relevant roles, we match requester’s security attributes against conditions required for corresponding role.

Role security policy is a set \( \text{rsp} = (r, d, p, c) \) where:
- \( r \) is a role identifier;
- \( d \) is the domain name the role belongs to;
- \( p = \{P_1:s_1, ..., P_n:s_n\} \) is a subset where \( P_i \) is the parameter of a privilege assigned to the role, \( s_i \) is the service identifier that associate with corresponding privilege;
- \( c = \{C_1:c_1, ..., C_n:c_n\} \) is a subset where \( C_i \) is the parameter of a conditions/requirements must be met to become the role, \( c_i \) is the value assigned to it.

**Example:**

\( \text{rsp} = (\text{Physician}, \text{ABC medical centre}, \{\text{read: patient health record; update: patient health record; forward: patient health record}\}, \{\text{ID: driver's license number; professional number: physician registration number}\}) \)

This role security policy states as: to play a role of physician in ABC medical centre, requester needs to provide valid driver’s license number and a physician registration number; a physician role has privilege to read, update and forward patient health records.

**Definition 2 Service Security Policy**

This policy specifies which roles have access right to the service and the conditions (provisions or obligations) associated to the specific privileges type on the service access.

Service security policy \( \text{ssp} = (s, r, a) \) where:
- \( s \) is service identifier;
- \( r = (r_1, r_2, ..., r_n) \) is role identifier, it indicates which roles can access the service;
- \( a = \{A_1:P_1, ..., A_n:P_n\} \) is a subset where \( A_i \) is the condition identifier, which means the correlating condition must be met to be able to access the service, \( P_i \) is the corresponding privileges associate with the condition.

**Example:**

\( \text{ssp} = (\text{Patient health records}, (\text{Physician, Nurse, Surgeon}), \{\text{privacy policy123:forward}\}) \)

This is an example of service security policy for Patient Health Records, only physician, nurse or surgeon is
allowed to access the records, and access can only proceed after the eligible requester agreed with provisions or obligations stated in the policy.

**Definition 3 Organization General Security Policy**

These policies specify organization level general security requirements, which are security requirements all the requests must be met to access any services provided by the organization. Organization general security policy $dsp = (d, dp_1, dp_2, \ldots, dp_n)$ where:

- $d$ is organization/domain name;
- $dp = \{DP_1; dp_1, \ldots, DP_n; dp_n\}$ is a subset where: $DP_i$ is the parameter of a organization/domain level security policy, $dp_i$ is the value assigned to it;

**Example:**

$dsp = (ABC$ medical centre, {Digital certificate: X.509})

This example depicts ABC medical centre has an organization level general security policy: anyone wants to access ABC must have a valid X.509 certificate.

**Definition 4 Service Access Request**

Service access request specify elements of access request sent by requester that include security attributes of requester; organizational general security policy of requester when it applies and services he/she requires. The requester’s attributes are IDs, properties, or other credentials that could identify the requester.

Service access request is a set $sar = (sa, dp, s)$ where:

- $sa = \{SA_1; sa_1, \ldots, SA_n; sa_n\}$ is a set where: $SA_i$ is the parameter of requester’s attributes (properties), $sa_i$ is the value assigned to it;
- $dp = \{DP_1; dp_1, \ldots, DP_n; dp_n\}$ is a set where: $DP_i$ is the parameter of a requester’s organization/domain level security policy, $dp_i$ is the value assigned to it;
- $s = \{S_1; P_1, \ldots, S_n; P_n\}$ is a set where: $S_i$ is service identifier that requester wants to access; $P_i$ is parameter of privileges the requester wants to perform on the service.

**Example:**

$sar = \{(ID: passport, driver’s license; professional membership: physician registration number}, \{Digital certificate: X.509\}, \{Patient Health Record: read\})$

The requester in example shows that requester discloses his/her passport, drivers’ license and physician registration number in order to access Patient Health Record, and the requester’s domain also use X.509 certificate.

### 6.2. Formalization of Security Control Rules

In this section we specify the rules for security access checking. Security control works as follows: the requester must qualify for certain roles that have requested privileges on requested service; the requester must agree with provisions or obligation associate with specific access actions of required service if there is any.

**Definition 5 Security Control Rule 1**

There are two steps to go through for all access requests:

- First step in security control is to compare service request with organization general security policy, and compare requester attributes with role requirements that has privileges on requested service.

Let $sar = (sa, dp, s)$ be a service request, $dsp = (d, dp)$ be a general security policy of requested service and $rsp = (r, d, p, c)$ be a role security policy in the organization. A role will be assigned to the requester, requester can pass first phase security control if following statements are true:

- For $\forall sar.dp.DP_i, i = 1, \ldots, n, \exists k, k = 1, \ldots, n$:
  
  $sar.dp.DP_i = dsp.dp.DP_k \land sar.dp.dp_i = dsp.dp.dp_k$;

- For $\forall sar.sa.SA_i, i = 1, \ldots, n, \exists k, k = 1, \ldots, n$:
  
  $sar.sa.SA_i \supseteq rsp.c.C_k \land sar.sa.sa_i = rsp.c.c_k$.

As defined above, requester must meet the security requirements of $dsp$ (organization level) and $rsp$ (role level), $sar$ must match $dsp$; attributes disclosed by requester $sar.sa$ need at least include all the requirements from role conditions $rsp.c$ and the value of these attributes must match value of $rsp.c$, but not limit on that. Once these requirements are all satisfied, requester can proceed to next step: service security control.

**Definition 6 Security Control Rule 2**

Let $r$ be a role identifier assigned to requester, $sar = (sa, dp, s)$ be service request, $ssp = (s, r, a)$ be security policies of the service the requester wants to access. The request is finally accepted when following statements are true:

- $sar.s = ssp.s$;
- For $i = 1, \ldots, n, r \in ssp.ri$;
- If $ssp.a \neq \{\emptyset\}$
  
  - $\forall sar.s.P_i, i = 1, \ldots, n, \exists k, k = 1, \ldots, n$;
  
  - $\forall ssp.a.a_i, i = 1, \ldots, n$
    
    - $r$ agreed with $a_i$. Where $sar.s.P_i \subseteq ssp.a.a_i$.

This is specific to each targeted service which the requester has gain access right for. A comparison is performed: first we compare role $r$ gained by requester to roles group defined in service; Secondly, we check
whether there are any conditions associate with the desired privileges of requester on the service, if so, the requester will be asked to agree on provisions or obligations associated with the access behavior.

7. Related Work

There are some related studies on security polices in Web Services. [4] Presented an access model (Ws-AC) for web services, it provided a simple logic model to represent policies and requests which are valuable for our paper. But it does not look at policy interactions in the context of service composition and cross-organization collaboration.

In [5], author proposed a WS-Broker to perform some tasks that are similar to those of the Policy Controller. It is a centralized service that provides services for all the participating organizations. While in our work we relate Policy Controller to each service provided by the participating organizations. Only organizations hold all information about how they provide services. However this information is critical for generating integrated security policies from different participants. Our Policy Controller resides inside the organization could overcome this problem.

None of these studies go into the depth of security issues in cross-organization collaboration. Different types of collaboration will raise different interaction problems that can affect security policies. Our goal is thus to provide a first step to identify types of cross-organization collaborations in this context; define policy integration rules; propose a framework that could identify underlying collaboration types, generate integrated security requirements users. Most importantly our work is in the context of business collaboration which involves multiple organizations rather than simple interaction between individual requester and single service provider.

8. Conclusions and Future work

In the collaborative business world, controlled access is a great concern. The peer based nature and emerged situations in business collaboration make the issue even more complicated. Before mechanism and tools can be developed for security policy checking, negotiation and enforcement, we believe it is critically important to comprehend the requirements of authorization issues in the context of business collaboration. In this paper, we discuss all the necessary elements in specifying security policies, identify different collaborations patterns, present rules for generating integrated security policies. We have proposed a framework for handling authorization control for business collaboration. The work presented here is only at its initial stage. However we believe it has laid a good foundation for future work in the area of policy consistency checking, policy negotiation, and security policy enforcement.

9. References