5-1-2013

Secure access control for health information sharing systems

Suhair Alshehri
Rochester Institute of Technology

Rajendra K. Raj
Rochester Institute of Technology

Recommended Citation
S. Alshehri and R. K. Raj, "Secure Access Control for Health Information Sharing Systems," 2013 IEEE International Conference on Healthcare Informatics, Philadelphia, PA, 2013, pp. 277-286. doi: 10.1109/ICHI.2013.40

Secure Access Control for Health Information Sharing Systems

Suhair Alshehri and Rajendra K. Raj

Department of Computer Science
B. Thomas Golisano College of Computing & Information Sciences
Rochester Institute of Technology
Rochester, New York 14623, USA
sxa3788@rit.edu, rkr@cs.rit.edu

Technical Report
May 2013

Abstract. The 2009 Health Information Technology for Economic and Clinical Health Act (HITECH) encourages healthcare providers to share information to improve healthcare quality at reduced cost. Such information sharing, however, raises security and privacy concerns that require appropriate access control mechanisms to ensure Health Insurance Portability and Accountability Act (HIPAA) compliance. Current approaches such as Role-Based Access Control (RBAC) and its variants, and newer approaches such as Attribute-Based Access Control (ABAC) are inadequate. RBAC provides simple administration of access control and user permission review, but demands complex initial role engineering and makes access control inflexible. ABAC, on the other hand, simplifies initial setup but increases the complexity of managing privileges and user permissions. These limitations have motivated research into the development of newer access control models that use attributes and policies while preserving RBAC's strengths. The BiLayer Access Control (BLAC) model is a two-step method being proposed to integrate attributes with roles: an access request is checked against *pseudoroles*, i.e., the list of subject attributes (first layer), and then against rules within the policies (second layer) associated with the requested object. This paper motivates the BLAC approach, outlines the BLAC model, and illustrates its usefulness to healthcare information sharing environments.

1 Introduction

Secure information sharing has become crucial across distributed settings in many application domains, especially in healthcare where the Health Information Technology for Economic and Clinical Health Act (HITECH) of 2009 provides "meaningful use" incentives for sharing Electronic Health Records (EHRs) among healthcare providers. Such sharing within or across healthcare organizations has been shown to provide optimal yet cost-effective healthcare services [1]. However, such sharing also raises serious concerns about the security and privacy of these records.

In the US. the Health Insurance Portability and Accountability Act (HIPAA) specifies strict protection requirements for identifiable healthcare information. One major HIPAA requirement is the use of appropriate access control mechanisms, with Role-Based Access Control (RBAC) [2] and its variants are currently used, and newer approaches such as Attribute-Based Access Control (ABAC) [3] have been proposed. However, both RBAC and ABAC have inadequacies.

In RBAC, access to objects is based on the subjects' (users and subjects are used synonymously in the literature, and we use either term depending on the context) roles, i.e., their job functions, such as "internist," "cardiologist" and "neurologist." Although RBAC provides simpler access administration and user permission review, and is widely deployed [4], it has several drawbacks. First, RBAC requires an expensive process to define and structure roles known as "role engineering." Second, due to RBAC's coarse-granularity, improper information disclosure and modification attacks by insiders are not uncommon [5–9] because roles–as typically defined–are not sufficiently granular to restrict data access only to the "proper" personnel. For instance, a patient record authorized for internists in a medical facility may make it accessible to *all* internists, even to those internists not involved with the patient's care in violation of HIPAA's Privacy Rule. Third, RBAC typically supports only predefined and static policies, making it inflexible in dynamically changing environments [10].

In ABAC [11], each subject's specific attributes are used to define access policies, with object access requests allowed or denied by checking against these attributes within these policies. The internists' example discussed above is not problematic because the ABAC access policy will include the specific identifying attributes of internists involved in a patient's care, thus automatically denying access to other internists. The use of attributes helps ABAC support fine-grained access control policies that can be evaluated dynamically in real-time, and are flexible enough to support environments with frequently changing user permissions. Despite these undeniable strengths, ABAC has two problems. First, ABAC is more complex due to the large numbers of rules that needed to be checked for access decisions; for n attributes, ABAC may require 2^n possible rules. Second, management of privileges, user permissions, and permission review for a specific user suffer from poor performance as large sets of rules must be executed [10].

Although RBAC and ABAC have their strengths, their limitations have led to a NIST call [10] for the development of a policy-enhanced RBAC model, which incorporates attributes while maintaining RBAC's advantages. Three approaches have been identified, but they have their own limitations. First, the *dynamic roles* approach, which uses attributes to assign roles to subjects, has been studied by Al Kahtani et al. [12] and Huang et al. [13]; however, the drawbacks of this approach include RBAC limitations such as the lack of granularity and lack of dynamic adaptability. Second, the *attribute-centric* approach defines roles as another attribute of subjects, thereby inheriting the disadvantages of ABAC without any of the advantages of RBAC. Third, the *role-centric* approach uses attributes to constrain permissions assigned to roles, and initially seems appealing because attributes can help limit unrestricted access provided by coarse-grained roles [14]; however, it does not permit full use of ABAC's flexibility. These approaches, which are further elaborated in Section 4, all have drawbacks.

The fundamental research question this paper attempts to answer is whether it is possible to develop an approach that combines the best of RBAC and ABAC with fewer drawbacks, and is appropriate for health information sharing environments. As a result, the BiLayer Access Control (BLAC) model was developed to support the use of attributes while preserving the advantages of RBAC. The main contribution of the paper is the development of the BLAC model and showing that it works for access control in the healthcare domain.

The rest of this section summarizes the necessary background in access control models to pave the way to the discussion of the BLAC model in Section 2. Section 3 demonstrates the applicability of the BLAC model to healthcare information sharing environments. Related work is discussed in section 4, and the current status and future work is presented in section 5.

1.1 Background

We briefly review RBAC [15] and ABAC [3] here; refer to the cited papers for more information about these models.

RBAC. In the core RBAC model [15] included in the NIST standard, RBAC is defined in terms of five basic data sets: subjects, roles, objects, operations, and permissions, as depicted in Figure 1.

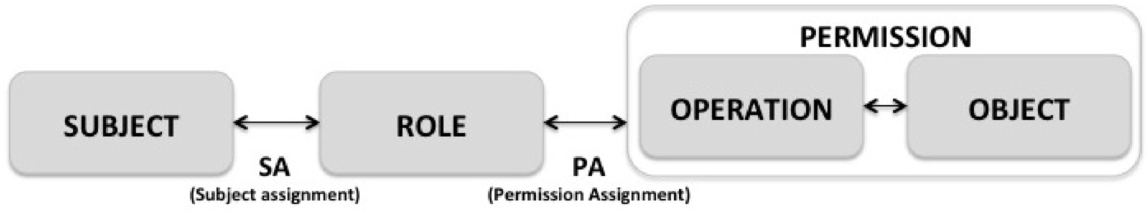


Fig. 1. The Basic Components of RBAC

Subjects are assigned to roles, and roles are associated with permissions that define which operations can be performed over which objects. Subjects are defined individuals, and roles represent job functions within the organization. In the context of healthcare, roles could be “doctor,” “nurse,” and “staff.” Operations can be “read a record,” “add a record,” and “modify a record.”

ABAC. Unlike RBAC, ABAC has not yet been standardized, but its fundamental concept is the use of attributes to define access rules [3]. ABAC is defined in terms of three basic components: subjects, environments, and objects. In addition, attributes are sets of key-value pairs to describe subjects, objects, and environments. The basic components of ABAC is shown in Figure 2.

For example, subject attributes could be $<Provider, Doctor>$ and $<Department, Cardiology>$; object attributes could be $<PatientName, Bob>$ and $<DocumentType, SummaryOfCare>$; and an environment attribute could be $<AccessIP = 192.123.*.*>$. Here, an example access rule could be *A doctor who works in Cardiology can access Bob’s summary of care report from a computer on a specific subnet of the hospital’s network.*

Attributes can be static or dynamic, depending on how frequently their values change. Access rules, i.e. policies, are defined using attributes. Requesters must possess attributes that satisfy relevant rules to be granted access to requested objects.

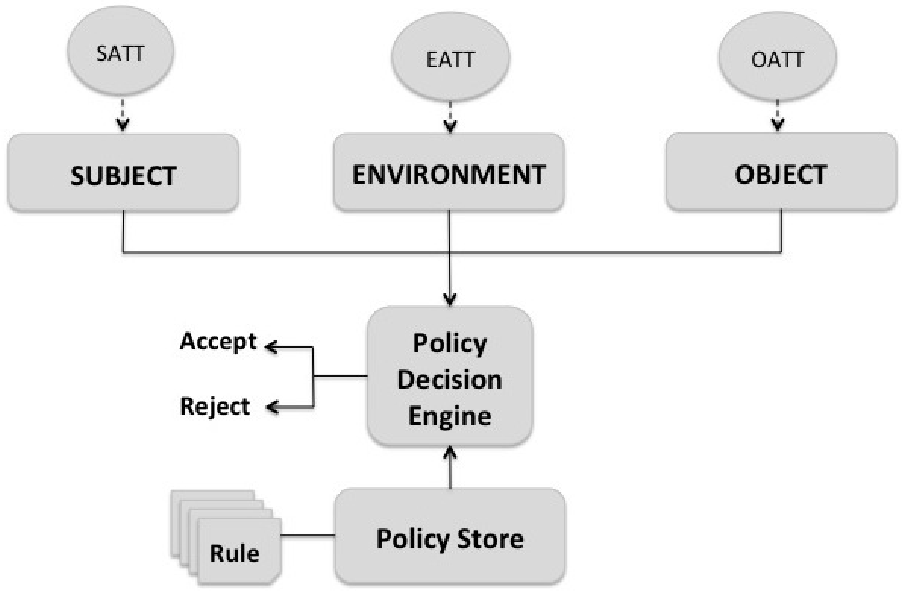


Fig. 2. The Basic Components of ABAC

2 An Overview of the BLAC Model

The BLAC model was designed to support attributes and policies while preserving the advantages of RBAC. The model uses the concept of *pseudorole*, which is defined as a set of static attributes of subjects. A pseudorole is not a real role, which is traditionally defined as a job function. Subjects' attributes are categorized as *static* (when attribute values typically do not change) and *dynamic* (when attribute values change frequently): static attributes are used to generate pseudoroles, and static and dynamic attributes are used in policies to constrain pseudoroles.

This section describes the BLAC model, discusses the generation of pseudoroles, and then identifies and uses a set of quality characteristics to compares the BLAC approach with others being currently studied.

2.1 The BLAC Model

In BLAC, subjects are associated with pseudoroles, which are composed of a set of static attributes, and objects are associated with policies, which specify how attributes are to be considered for access requests. When an access request is made, the policy associated with the requested object is first checked to see whether the requester has the required pseudorole or not. If the requester holds the right pseudorole, rules within the policies are then checked for additional constraints to approve or deny the access request. This two-step process, which inspired the name BiLayer Access Control, permits fine-grained access decisions. Figure 3 shows the access requests evaluation flow in BLAC model.

BLAC is formally described using the following tuple:

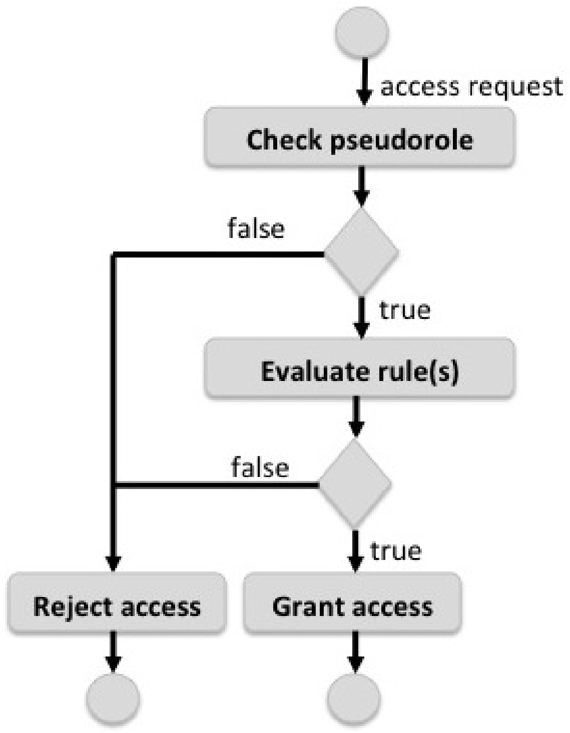


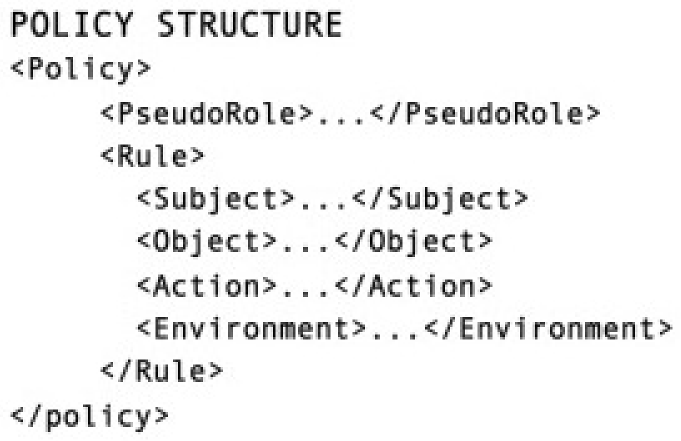
Fig. 3. Access Request Evaluation Flow in the BLAC Model

$$M = (S, O, E, A, PR, P, SPR, OP)$$

Here, S is Subject, O is Object, E is Environment, A is Action, PR is pseudorole, P is Policy, SPR is subject-pseudorole assignment relation, and OP is object-policy assignment relation. More details are provided below:

- **S:** is a set of subjects (users) that need access to objects. Subjects are associated with a predefined finite set of attributes $SATT$. Each attribute is represented by a name, along with its associated value. Examples of subject attributes could be “provider,” “department,” and “location.” For a given subject, these attributes could be respectively associated with values such as “physician,” “OB/GYN,” and “hospital A.”
- **O:** is a set of objects that are accessed by subjects. Objects are also associated with a predefined finite set of attributes $OATT$. Examples of object attributes in the healthcare domain could be “patient name” and “medical record number;” appropriate values would be associated with each attribute.
- **E:** is the environment, which describes the access context under which access may be provided. Environments are associated with a predefined finite set of attributes $EATT$. Examples of environment attributes could be “access time” and “system mode.”

- **A:** is a set of actions, which subjects request, to be performed over objects. Actions are also associated with a predefined finite set of attributes $AATT$. Examples of action attributes are "read" and "modify."
- **PR:** is a set of pseudoroles that are composed of n attributes. Each subject is assigned to a pseudorole for initial coarse-grained access control. The process for generating pseudoroles is described in Section 2.2.
- **P:** is a set of access policies for fine-grained access control. A policy consists of two elements: a Boolean function called PseudoRole, and a set of zero or more rules. Each rule has four sub-elements also defined as a Boolean function specifying the range of values that must be satisfied for the Subject, Object, Action, and Environment attributes. A policy structure is shown in Figure 4, following XACML policy format [16].



```
POLICY STRUCTURE
<Policy>
     <PseudoRole>...</PseudoRole>
     <Rule>
       <Subject>...</Subject>
       <Object>...</Object>
       <Action>...</Action>
       <Environment>...</Environment>
     </Rule>
</policy>
```

Fig. 4. Policy Structure

The PseudoRole is used to do an initial check whether a subject requesting access to an object holds the needed pseudorole specified by the policy, otherwise the access request is rejected. Next, each rule is checked to see if the access conforms to the specified values for subject, object, action, and environment attributes, otherwise access is denied.

- **SPR:** is the subject-pseudorole assignment relation that is a one-to-many mapping from pseudoroles to subjects.
- **OP:** is the object-policy assignment relation that is a one-to-many mapping from policies to objects.

We can now define BLAC formally.

Definition 1. *BLAC.*

- *S, O, E, A, PR, P are subjects, objects, environments, actions, pseudoroles, and policies, respectively.*
- $SATT_k$ *(*$1 \leq k \leq K$*),* $OATT_m$ *(*$1 \leq m \leq M$*), and* $EATT_n$ *(*$1 \leq n \leq N$*) are the predefined finite set of attributes for subjects, objects, and environments, respectively.*

- *assigned-attributes(s), assigned-attributes(o), and assigned-attributes(e) are attribute assignment relations for subject s, object o, and environment e, respectively:*
  assigned-attributes(s) $\subseteq SATT_1 \times SATT_2 \times \ldots \times SATT_K$
  assigned-attributes(o) $\subseteq OATT_1 \times OATT_2 \times \ldots \times OATT_M$
  assigned-attributes(e) $\subseteq EATT_1 \times EATT_2 \times \ldots \times EATT_N$.
- PR_i *(*$1 \leq i \leq I$*), is a set of* $g \in SATT$ *such that* $g \leq K$.
- $SPR \subseteq S \times PR$*, a subject-pseudorole assignment relation that is a one-to-many mapping from pseudoroles to subjects.*
- *assigned-subjects(pr)* $= \{s \in S \mid (s,pr) \in SPR\}$*, the mapping of a pseudorole pr onto a set of subjects.*
- P_g *(*$1 \leq g \leq G$*), is a set of access policies that determines whether a subject s can access an object o, such that,* $P_g = \{FPR, FR_1, \ldots, FR_n\}$*, is a policy that consists of a Boolean function of PR, and a finite set of Boolean functions of access rules of s, o, a, and e's attributes.*

Figure 5 shows the components of the BLAC model.

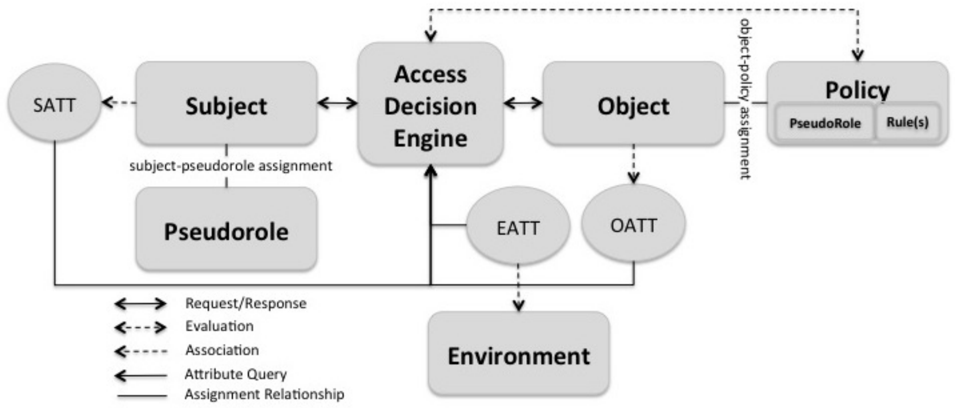


Fig. 5. The Components of the BLAC Model

2.2 Pseudorole Generation

In BLAC, pseudoroles need to be generated from static attributes of subjects as described in the following paragraphs. The approach of pseudorole generation is different from role mining approaches [17] in RBAC systems, which use permissions possessed by all subjects to generate roles. We use the values of the attributes associated with all subjects to generate pseudoroles. The following paragraphs describe our approach.

First, we find all distinct values in the first attribute column. Each value will be considered as a root of a tree graph. We will then find all distinct values of the second attribute column and add them as nodes to each rooted tree. We will keep adding new nodes to new levels for each tree depending on the number of attributes used to generate pseudoroles. Thus, if three attributes are used, this process will be repeated three times. The formal algorithm used for generating pseudoroles is not presented here.

After creating the trees, each path from a leaf to the root will be consider as a pseudorole. For example, in Figure 6, we have four trees with 24 distinct pseudoroles. In the first tree, the path from the leaf to the root is “A,” “OB/GYN” and “physician.” Thus, “Physician OB/GYN A” is a pseudorole.

We could consider all the generated pseudoroles in a system even though not all the pseudoroles can currently be associated with subjects because we may later add new subjects to these unused pseudoroles. However, we could eliminate unreasonable pseudoroles such as “Physician Billing A” and “Physician Billing B.” At the end, we need to take into account that though “Physician OB/GYN A” is a pseudorole, it is still three distinct attribute values.

2.3 Comparing the BLAC Model with Other Approaches

As stated, the BLAC model was proposed to make effective use of attributes in access control while preserving RBAC’s advantages. To compare BLAC with other approaches including RBAC, ABAC, the NIST approaches, ABP-RBAC [13] and RABAC [14], an appropriate set of quality characteristics was identified, as shown in Table 1. The comparison omits the dynamic roles approach (as it is subsumed by the approach by ABP-RBAC [13]), the role centric approach (subsumed by RABAC), and the attribute centric approach (subsumed by the basic ABAC approach).

Flexibility. RBAC requires the use of predefined roles and permissions that are associated with these roles, thus, it is a static model that cannot dynamically adopt to frequent changes in access privileges and subject permissions. ABP-RBAC is also a static model because the attributes are only used to generate the roles that will be used for making access decisions. ABAC, on the other hand, supports dynamic environments as it processes access requests during run-time according to requesters’ attributes. RABAC uses attributes to constrain the predefined permissions assigned to roles for each activated sessions, thus, it is considered a static model. BLAC, however, associate subjects with pseudoroles, but it does not assign fixed permissions to these pseudoroles as in RBAC. To preserve the needed flexibility, it instead utilizes policies that are compared, at the time of access requests, initially against pseudoroles and then against rules that are composed of subject, object, action, and environment attributes for dynamic access control.

Granularity. RBAC provides coarse-grained access control that is inadequate for data-sharing environments as access rules must be applied at a low-level of granularity such as a single file. In larger systems that require finer permissions, more separate roles are defined to capture the different sets of permissions which can result in “role explosion.” ABP-RBAC inherits the lack of granularity

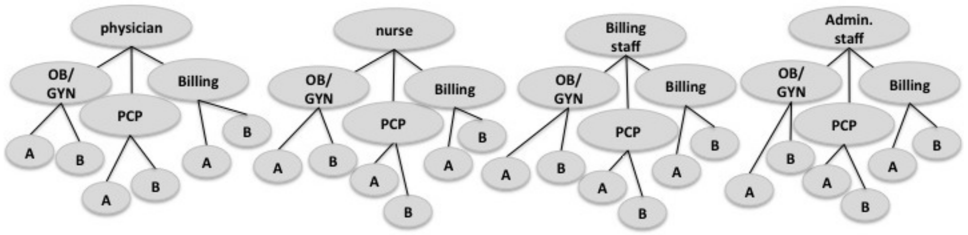


Fig. 6. Pseudorole Generation in the BLAC Model

Table 1. Quality Characteristics

Characteristic	**Description**
Flexibility	the ability to support frequent changes in policy
Granularity	the granular level at which permissions can be applied over objects
Authorization Complexity	a measure of the complexity for access control models to make authorization decisions
Privilege Modifiability	the ability to modify access privileges
Permission Modifiability	the ability to modify rights of subjects to access objects
Revocability	the capability to revoke rights of subjects to access objects
Permission Reviewability	the ability to determine the set of available permissions for a particular subject
Setup Complexity	the complexity of the required initial setup

because it uses roles only for making access decisions. ABAC , on the other hand, supports fine-grained access control because it restricts access based on the attributes of subjects, objects and environments, and it defines permissions according to fine-grained access rules. RABAC uses the attributes to constrain the maximum permissions available in each session that are determined by the activated roles, thus, it provides fine-grained access control. BLAC, on the other hand, supports two levels of granularity: a high-level and a low level to accommodate the requirements of information sharing environments. The high level of granularity is represented by the pseudoroles, and the low level of granularity is represented by the use of fine-grained policies.

Authorization Complexity. RBAC makes authorization decisions in advance as it associates a set of permissions to roles. Thus, when a subject requests an operation over an object, RBAC checks what role is assigned with this subject, and if the requested permission (the operation over the object) is associated with the subject's role, the access is granted; otherwise, the access is denied. ABP-RBAC follows the same approach as RBAC though the subject-role and role-permission assignment operations uses attribute-based policies. ABAC makes access decisions at the time of requests as it compares the attributes associated with the requester against all applicable policies. If the policies are satisfied with the subject attributes, the access is granted; otherwise, the access is denied. When a subject makes an access request, ABAC policy engine must check each policy in the policy store if it applies to the request. Finally, all applicable policies will be evaluated. RABAC also evaluates access requests at run-time to find final available permissions based on permission filtering policies. BLAC also makes access decisions at the time of requests. When a subject requests an access, BLAC first checks the pseudoroles associated with the subject, and then compares it with the pseudooles included in the policy associated with the requested object, not each policy in the policy store as in ABAC. If the policy is satisfied with the subject's pseudorole, BLAC further check the rules within the policy. If one of the rules returns true, then the access is granted; otherwise, the access is denied.

Privilege Modifiability. RBAC handles changes in privileges by modifying and updating the sets of permissions associated with roles without changing privileges for each subject in the system, thus RBAC simplifies the administration of privileges. ABP-RBAC and RABAC inherits this simplicity, in the former, by modifying the sets of permissions associated with roles, and in the latter, by modifying the filtering policies and/or the permissions associated with the roles. ABAC, on the other hand, copes with changes in privileges by modifying or revoking all relevant policies in the policy store. ABAC thus complicates the management of privileges. BLAC handles changes in privileges by updating or revoking the policies that are associated with the relevant objects. However, the modification of privileges in BLAC is a complex process that requires security administrators to intervene before policies are changed.

Permission Modifiability. RBAC handles the change of subject permissions by modifying subject membership into roles. When subject permissions change, the subject is revoked from the current role and a new subject membership can be established based on the new access requirements. ABP-RBAC and RABAC handle the change of subject permissions similar to RBAC. ABAC, on the other hand, complicates management of subject permissions because it is hard to map changes in subject permissions to subjects attributes. However, relevant policies must be updated or revoked to accommodate the current requirements. BLAC handles the change of subject permissions very similar to ABAC. All relevant policies must be updated or revoked to accommodate the new requirements. Thus, the modification of subject permissions in BLAC is still a complex process.

Revocability. RBAC handles subject revocation by modifying subject membership into roles. When a subject is revoked, the subject is simply revoked from all the assigned roles. ABP-RBAC and

RABAC also handle subject revocation in a similar manner to RBAC. ABAC complicates subject revocation because relevant policies must be revoked to accommodate the new requirements. BLAC, however, handles subject revocation by simply revoking the subject from the assigned pseudorole, thus subject revocation is simpler in BLAC.

Permission Reviewability. RBAC simplifies reviewing the available set of permissions assigned to a single subject. Permission review can be achieved by checking the set of permissions associated with the roles assigned to the subject. ABP-RBAC and RABAC also handle permission review similar to RBAC. ABAC complicates the permission review because a large set of policies must be checked to determine the available set of permissions for a subject. Though BLAC performs the process of permission review very similar to ABAC, it does not execute all the policies to determine the available set of permissions for a single subject. BLAC simplifies the process by only reviewing a set of policies where the pseudoroles within these policies match the pseudorole of the subject.

Setup Complexity. Both RBAC and RABAC suffer from a complex up-front process called role engineering, which is a process for developing an RBAC role structure for a system. ABP-RBAC, on the other hand, overcomes the problem of role engineering by automating the assignment process. ABAC simplifies the set up as it directly utilizes attributes. Though BLAC uses pseudoroles for initial coarse-grained access control, it does not need to determine a role structure as in RBAC. This is because, the pseudoroles are derived from subjects attributes. However, subject-pseudorole assignment and object-policy assignment functions are still needed but they can be simply implemented. Policies, on the other hand, must be created by security administrators. Setup in BLAC would be further simplified by methods for automatic creation of such policies from object attributes and subjects that create the objects.

The above comparative analysis of the various access control approaches is summarized in Table 2.

Table 2. Comparison Summary of Quality Characteristics of RBAC, ABAC, ABP-RBAC [13], RABAC [14], and BLAC

Characteristic	**RBAC**	**ABAC**	**ABP-RBAC**	**RABAC** al.	**BLAC**
Flexibility	Low	High	Low	Low	High
Granularity	Low	High	Low	High	High
Authorization Complexity	Static	Dynamic	Static	Dynamic	Dynamic
Privilege Modifiability	Simple	Complex	Simple	Simple	Complex
Permission Modifiability	Simple	Complex	Simple	Simple	Complex
Revocability	Simple	Complex	Simple	Simple	Simple
Permission Reviewability	Simple	Complex	Simple	Simple	Simple
Setup Complexity	Complex	Simple	Simple	Complex	Simple

3 Using BLAC in Healthcare

To show the BLAC model's applicability to healthcare information sharing, we first develop a *use case* based on a medical center with two hospital affiliates, hospital A and hospital B. Here, nine subjects make use of an EHR system for managing patients' records in both hospitals. Each subject is defined by *name*, identification number (*ID*), *gender*, the field of the healthcare *provider* their *department*, and their office *location*. The nine subjects are described in Table 3:

Table 3. Subjects' Attributes

Name	**ID**	**Gender**	**Provider**	**Department**	**Location**
E. Robert	345-765	Female	Physician	OB/GYN	A
A. Mark	526-874	Male	Physician	OB/GYN	A
H. John	231-938	Female	Nurse	OB/GYN	A
M. Martin	657-923	Female	Administrative Staff	OB/GYN	A
E. Arthur	112-681	Male	Billing Staff	Billing	A
J. Fox	437-348	Male	Physician	PCP	B
H. Anderson	256-828	Female	Nurse	PCP	B
F. Brown	562-910	Female	Administrative Staff	PCP	B
D. Lee	102-581	Male	Billing Staff	Billing	B

Health records are defined in the system by patient name, patient MRN, patient DOB, and the ID of the physician responsible for treating the patient. Health records are split into three sections: (1) demographical, (2) clinical, and (3) billing.

To preserve the privacy of patient records, access control rules are defined in this use case as follow:

1. Physicians and nurses are allowed to read and modify the demographical and clinical sections for patients who are under their responsibility in normal situations, with the exception of psychiatric data.
2. Physicians and nurses are allowed to read and modify the demographical and clinical sections for non-patients in emergency situations, with the exception of psychiatric data.
3. Subjects are not allowed to delete records in any section.
4. Administrative staff are allowed to read and modify records within the demographical section when they are on duty.
5. Billing staff are allowed to read and modify records within billing section, and only read records within demographical section when they are on duty.

Understanding subject, object, and environment attributes. Subjects are the healthcare providers using the EHR system; from Table 3, there are nine subjects and each is defined with six attributes: name, ID, gender, provider, department, and location. Objects are the health data that these subjects add and modify, and they are identified by four attributes: name, MRN, DOB, and doctorID. Environment attributes specify the access context for making access decisions, and the environment is defined by two attributes: access-time (when requesters perform access requests) and mode (normal or emergency situation).

Identification of pseudoroles. To identify the possible set of attributes that could be used for generating pseudoroles, the system needs to know which are the subject attributes and also distinguish static attributes from dynamic attributes. In this use case, "Name," "ID," "Gender," "Provider,"

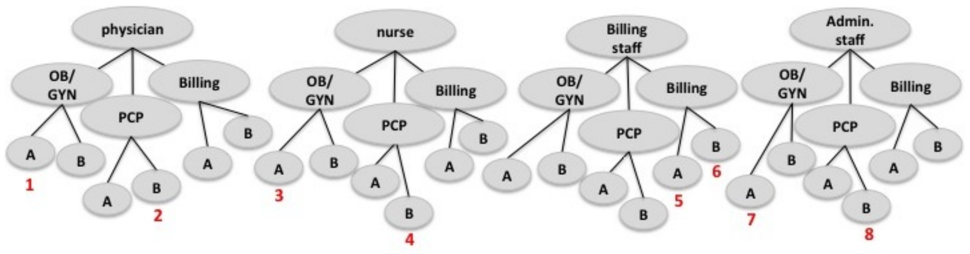
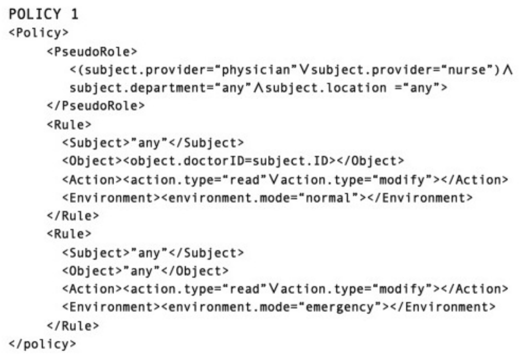
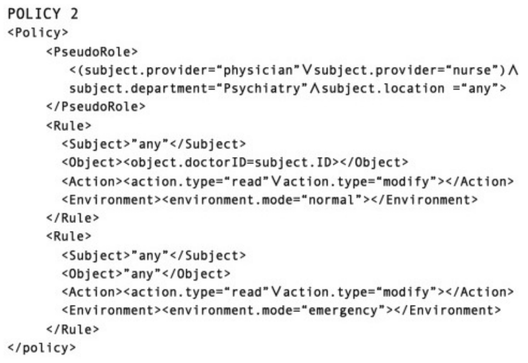


Fig. 7. Generating Pseudoroles for the Healthcare Use Case



```
POLICY 1
<Policy>
      <PseudoRole>
         <(subject.provider="physician"∨subject.provider="nurse")∧
         subject.department="any"∧subject.location ="any">
      </PseudoRole>
      <Rule>
        <Subject>"any"</Subject>
        <Object><object.doctorID=subject.ID></Object>
        <Action><action.type="read"∨action.type="modify"></Action>
        <Environment><environment.mode="normal"></Environment>
      </Rule>
      <Rule>
        <Subject>"any"</Subject>
        <Object>"any"</Object>
        <Action><action.type="read"∨action.type="modify"></Action>
        <Environment><environment.mode="emergency"></Environment>
      </Rule>
</policy>
```

Fig. 8. Policy 1



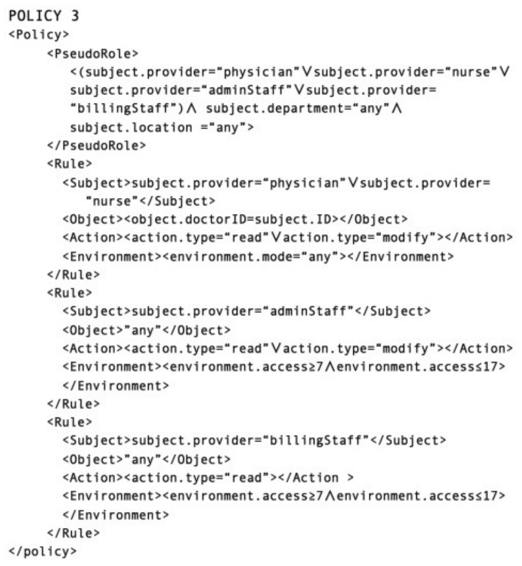
```
POLICY 2
<Policy>
      <PseudoRole>
         <(subject.provider="physician"∨subject.provider="nurse")∧
         subject.department="Psychiatry"∧subject.location ="any">
      </PseudoRole>
      <Rule>
        <Subject>"any"</Subject>
        <Object><object.doctorID=subject.ID></Object>
        <Action><action.type="read"∨action.type="modify"></Action>
        <Environment><environment.mode="normal"></Environment>
      </Rule>
      <Rule>
        <Subject>"any"</Subject>
        <Object>"any"</Object>
        <Action><action.type="read"∨action.type="modify"></Action>
        <Environment><environment.mode="emergency"></Environment>
      </Rule>
</policy>
```

Fig. 9. Policy 2

"Department" and "Location" are static, thus good candidates for generating pseudoroles. The system needs to ensure that dynamic attributes such as "age" are not used for generating pseudoroles as they change frequently. In our use case, "Name," "ID," and "Gender" are not used to generate pseudoroles, as they cannot minimize granularity, but "Provider," "Department," and "Location" can be used.

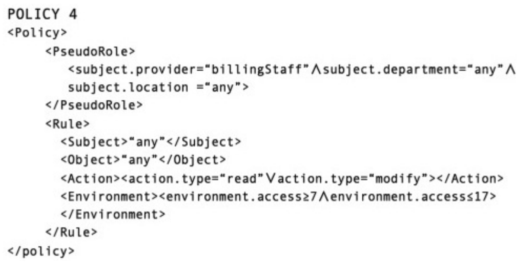
Generating pseudoroles. The BLAC approach for generating pseudoroles uses attributes whose values are associated with all subjects. Thus, if the number of total different attributes values is n, the number of possible pseudoroles is 2^n, but meaningful pseudoroles are a subset of these pseudoroles. Based on the use case, the complete set of generated pseudoroles is shown in Figure 7. The eight meaningful pseudoroles used here are (1) "Physician OB/GYN A," (2) "Physician PCP B," (3) "Nurse OB/GYN A," (4) "Nurse PCP B," (5) "Administrative Staff OB/GYN A," (6) "Administrative Staff PCP B," (7) "Billing Staff Billing A," and (8) "Billing Staff Billing B."

Development of Policies. Based on the previously mentioned access rules and the structure of the health records, defined policies could be based on the policy structure shown in Figure 4. To enforce the access rules, health records within the "clinical section" are associated with Policy 1 shown in Figure 8. However, the portion of the health records, within the "clinical section," related to psychiatric data are associated with the Policy 2 shown in Figure 9. Health records within the "demographical section" are associated with Policy 3 shown in Figure 10, and finally, health records within the "billing section" are associated with Policy 4 shown in Figure 11.



```
POLICY 3
<Policy>
     <PseudoRole>
        <(subject.provider="physician"∨subject.provider="nurse"∨
        subject.provider="adminStaff"∨subject.provider=
        "billingStaff")∧ subject.department="any"∧
        subject.location ="any">
     </PseudoRole>
     <Rule>
       <Subject>subject.provider="physician"∨subject.provider=
          "nurse"</Subject>
       <Object><object.doctorID=subject.ID></Object>
       <Action><action.type="read"∨action.type="modify"></Action>
       <Environment><environment.mode="any"></Environment>
     </Rule>
     <Rule>
       <Subject>subject.provider="adminStaff"</Subject>
       <Object>"any"</Object>
       <Action><action.type="read"∨action.type="modify"></Action>
       <Environment><environment.access≥7∧environment.access≤17>
       </Environment>
     </Rule>
     <Rule>
       <Subject>subject.provider="billingStaff"</Subject>
       <Object>"any"</Object>
       <Action><action.type="read"></Action >
       <Environment><environment.access≥7∧environment.access≤17>
       </Environment>
     </Rule>
</policy>
```

Fig. 10. Policy 3



```
POLICY 4
<Policy>
     <PseudoRole>
        <subject.provider="billingStaff"∧subject.department="any"∧
        subject.location ="any">
     </PseudoRole>
     <Rule>
       <Subject>"any"</Subject>
       <Object>"any"</Object>
       <Action><action.type="read"∨action.type="modify"></Action>
       <Environment><environment.access≥7∧environment.access≤17>
       </Environment>
     </Rule>
</policy>
```

Fig. 11. Policy 4

4 Related Work

This section reviews foundational and related work in the area of access control models.

RBAC and RBAC Extensions. RBAC was first introduced and modeled by Ferraiolo and Kuhn [18] in the early 1990s, and subsequently, frameworks for four RBAC models were proposed by Sandhu et al. [19]. Since then, RBAC has been widely investigated in a variety of application domains, particularly to overcome its two main limitations: lack of granularity and lack of dynamic adaptability.

Other early RBAC variants included Context-based Access Control (CBAC) [20], where the access request context supplements roles. In Temporal Role-Based Access Control (TRBAC) [21], permissions assigned to roles can be enabled only during specific fixed or variable temporal periods. Location-Based Access Control (LBAC) [22, 23] places constraints on users' locations to meet the increasing security requirements by mobile applications and users. For example, access might be allowed only when users are at a specific location within the organization's premises or at a location that satisfies required security properties.

Other major extensions include Task-Role-Based Access Control (T-RBAC) [24,25], where access decisions are based son roles and tasks that represent fundamental units of business activity in enterprise environment; and Privacy-Aware Role-Based Access Control (P-RBAC) [26, 27], where privacy policies are incorporated to protect access to private and sensitive data.

ABAC and ABAC Extensions. Although the notion of attributes and policies was introduced through the use of security credentials [28–30], to the best of our knowledge, Yuan and Tong [3] were the first to introduce an ABAC model in the context of web services based on subject, object, and environment attributes; although they described their model in terms of authorization architecture and policy formulation, their paper did not consider operations on resources and did

not provide implementation details of policy evaluation. Other researchers [31–35] have presented various models for ABAC for different applications and domains. Hai-bo and Fan [31] described an ABAC model for web services that evaluates access requests for web 'vservices based on requesters' attributes. Alipour et al. [32] extended Yuan and Tong's ABAC model for use in service-oriented architecture environments. Zhu and Smari [33] presented an ABAC model for use in collaboration environments. Although these efforts advanced ABAC models, they did not consider issues such as privilege management, user permissions, and user permission review.
Revising the NIST-RBAC Standard. In section 1, we discussed the NIST initiative [10] to include attributes in access decisions while preserving RBAC's benefits. Three possible combination strategies for integrating attributes into RBAC were identified: *dynamic roles*, *attribute-centric* and *role-centric*. We used the drawbacks with these approaches to motivate the BLAC model.

1. **Dynamic roles.** In this approach, attributes are used to assign roles to subjects dynamically, thus addressing the problem of assigning subjects to roles. Al-Kahtani et al. [12] introduced a rule-based RBAC model that uses organization-defined rules to automatically assign users to roles based on user attributes and constraints on roles. Huang et al. [13] presented an integrated model of RBAC and ABAC. They use the composition of two layers: *aboveground* and *underground* with the former constructing a traditional RBAC model extended with environment constraints, and the latter using attribute-based user-role assignment policies and role-permission assignment policies to automate user-role assignment and role-permission assignment functions for the aboveground level or the traditional RBAC model. Both efforts however have not addressed the problem of granularity and dynamic adaptability, which is the focus of our work.
2. **Attribute centric.** This approach defines roles as another attribute of subjects, this does not assign roles to a set of permissions as in RBAC. As most of the standard ABAC models presented above can be used to implement attribute-centric ABAC, this approach thus inherits the disadvantages of ABAC and cannot benefit from the advantages of RBAC.
3. **Role centric.** In this approach, attributes are used to constrain the permissions assigned to roles. Although this approach retains RBAC's ability to compute the maximum set of permissions for a single subject, it lacks ABAC's flexibility due to the constraints imposed by roles. Jin et al [14] present a role-centric model that incorporates RBAC's basic elements, along with subject attributes, object attributes, and a permission filtering policy (PFP). The PFP makes use of subject and object attributes to constrain the available set of permissions.

5 Current Status and Future Work

Ensuring security and privacy compliance with HIPAA while sharing healthcare information continues to be a challenge. Access control mechanisms are fundamental to achieve such assurance.

As a result of the NIST-initiative [10] to develop a policy-enhanced RBAC model that integrate attributes while maintaining RBAC's advantages, this paper proposed the BLAC model. In this model, subject attributes are classified into static and dynamic based on the frequency of changes made to the attributes' values. Static attributes are used to compose pseudoroles that are associated with subjects, and static and dynamic attributes are used to constrain pseudoroles through the use of policies that are associated with objects.

The BLAC model provides bilayer access control: when an access request is issued, the pseudorole is checked (first layer), and if the requester has the right pseudorole, the rules within the policy will be checked further for fine-grained constraints (second layer). The paper demonstrated the applicability of the BLAC model to healthcare information sharing environments. BLAC is currently being implemented and performance results will be presented in the future. In a related study, we plan to analyze the nature of security threats in the domain of health information sharing and verify whether the BLAC model will perform effectively and efficiently for all of the threats identified. Finally, we plan to extend the proposed approach to ensure patient privacy is protected when their identity attributes are used for making access decisions.

References

1. The American Health Information Management Association: Health Information Exchange (2013)
2. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. Computer **29**(2) (Feb 1996)
3. Yuan, E., Tong, J.: Attributed Based Access Control (ABAC) for Web Services. In: Proceedings of the IEEE International Conference on Web Services. ICWS '05, Washington, DC, USA, IEEE Computer Society (2005) 561–569
4. Healthcare Information and Management Systems Society: Health Information Exchanges: Similarities and Differences HIMSS HIE Common Practices Survey Results White Paper (January 2011) `http://www.himss.org/ASP/topics_FocusDynamic.asp?faid=148`.
5. iHealthBeat: Hospital workers fired for improper access of football players' ehrs. California HealthCare Foundation (Feb 2011)
6. Chickowski, E.: Healthcare unable to keep up with insider threats. Dark Reading (May 2012)
7. Vijayan, J.: Three fired for accessing records of tucson shooting victims. Computerworld (Jan 2011)
8. Roney, K.: Titus regional medical center nurse fired over hipaa violation. Beckers Hospital Review (Jan 2012)
9. News, C.: Doctor probed for improper health record access (Dec 2011)
10. Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding attributes to role-based access control. Computer **43**(6) (June 2010) 79–81
11. Stepien, B., Matwin, S., Felty, A.: Advantages of a Non-Technical XACML Notation in Role-Based Models. In: 9th Annual International Conference on Privacy, Security, and Trust, IEEE (2011)
12. Al-Kahtani, M.A., Sandhu, R.: A model for attribute-based user-role assignment. In: Proceedings of the 18th Annual Computer Security Applications Conference. ACSAC '02, Washington, DC, USA, IEEE Computer Society (2002) 353–
13. Huang, J., Nicol, D.M., Bobba, R., Huh, J.H.: A framework integrating attribute-based policies into role-based access control. In: Proceedings of the 17th ACM symposium on Access Control Models and Technologies. SACMAT '12, New York, NY, USA, ACM (2012) 187–196
14. Jin, X., Sandhu, R., Krishnan, R.: Rabac: role-centric attribute-based access control. In: Proceedings of the 6th international conference on Mathematical Methods, Models and Architectures for Computer Network Security: computer network security. MMM-ACNS'12, Berlin, Heidelberg, Springer-Verlag (2012) 84–96
15. Sandhu, R., Ferraiolo, D., Kuhn, R.: The nist model for role-based access control: towards a unified standard. In: Proceedings of the fifth ACM workshop on Role-based access control. RBAC '00, New York, NY, USA, ACM (2000) 47–63
16. The Organization for the Advancement of Structured Information Standards (OASIS): OASIS eXtensible Access Control Markup Language (XACML) (2013)
17. Xu, Z., Stoller, S.D.: Algorithms for mining meaningful roles. In: Proceedings of the 17th ACM symposium on Access Control Models and Technologies. SACMAT '12, New York, NY, USA, ACM (2012) 57–66

18. Ferraiolo, D.F., Kuhn, D.R.: Role-Based Access Controlsn. In: 15th National Computer Security Conference. (1992)
19. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. Computer **29**(2) (feb 1996) 38 –47
20. Bertino, E., Bonatti, P.A., Ferrari, E.: Trbac: a temporal role-based access control model. In: Proceedings of the fifth ACM workshop on Role-based access control. RBAC '00, New York, NY, USA, ACM (2000) 21–30
21. Joshi, J., Bertino, E., Latif, U., Ghafoor, A.: A generalized temporal role-based access control model. Knowledge and Data Engineering, IEEE Transactions on **17**(1) (jan. 2005) 4 – 23
22. Damiani, M.L., Bertino, E., Catania, B., Perlasca, P.: Geo-rbac: A spatially aware rbac. ACM Trans. Inf. Syst. Secur. **10**(1) (February 2007)
23. Ray, I., Toahchoodee, M.: A spatio-temporal role-based access control model. In: Proceedings of the 21st annual IFIP WG 11.3 working conference on Data and applications security, Berlin, Heidelberg, Springer-Verlag (2007) 211–226
24. Oh, S., Park, S.: Taskrole-based access control model. Information Systems **28**(6) (2003) 533 – 562
25. Sainan, L.: Task-role-based access control model and its implementation. In: Education Technology and Computer (ICETC), 2010 2nd International Conference on. Volume 3. (june 2010) V3–293 –V3–296
26. Ni, Q., Trombetta, A., Bertino, E., Lobo, J.: Privacy-aware role based access control. In: Proceedings of the 12th ACM symposium on Access control models and technologies. SACMAT '07, New York, NY, USA, ACM (2007) 41–50
27. Masoumzadeh, A., Joshi, J.B.: Purbac: Purpose-aware role-based access control. In: Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems. OTM '08, Berlin, Heidelberg, Springer-Verlag (2008) 1104–1121
28. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy. SP '96, Washington, DC, USA, IEEE Computer Society (1996) 164–
29. Park, J.S., Sandhu, R.: Smart certificates: Extending x.509 for secure attribute services on the web. In: Proceedings of 22nd National Information Systems Security Conference (NISSC. (1999) 18–21
30. Bonatti, P., Samarati, P.: Regulating service access and information release on the web. In: Proceedings of the 7th ACM conference on Computer and communications security. CCS '00, New York, NY, USA, ACM (2000) 134–143
31. Shen, H., Hong, F.: An attribute-based access control model for web services. In: Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on. (dec. 2006) 74 –79
32. Alipour, H., Sabbari, M., Nazemi, E.: A policy based access control model for web services. In: Internet Technology and Secured Transactions (ICITST), 2011 International Conference for. (dec. 2011) 472 –477
33. Zhu, J., Smari, W.: Attribute based access control and security for collaboration environments. In: Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National. (july 2008) 31 –35
34. Lang, B., Foster, I., Siebenlist, F., Ananthakrishnan, R., Freeman, T.: A flexible attribute based access control method for grid computing. Journal of Grid Computing **7** (2009) 169–180
35. Li, F., Weerasinghe, D., Patel, D., Rajarajan, M.: An user-centric attribute based access control model for ubuquitous environments. In Zhang, J., Wilkiewicz, J., Nahapetian, A., eds.: Mobile Computing, Applications, and Services. Volume 95 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg (2012) 361–367