Hindawi
Security and Communication Networks
Volume 2018, Article ID 1572812, 13 pages
https://doi.org/10.1155/2018/1572812

Research Article

Towards a Secure and Borderless Collaboration between Organizations: An Automated Enforcement Mechanism

Samira Haguouche and Zahi Jarir

LISI Laboratory, Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco

Correspondence should be addressed to Samira Haguouche; s.haguouche@uca.ma

Received 13 July 2018; Accepted 4 October 2018; Published 21 October 2018

Academic Editor: Kuo-Hui Yeh

During the last decade, organizations have been more and more aware of the benefits of engaging in collaborative activities. To attain a required collaborative objective, they are obligated to share sensitive resources such as data, services, and knowledge. However, sharing sensitive and private resources and exposing them for an external usage may prevent the organizations involved from collaborating. Therefore, this usage requires more preoccupation with security issues. Access control is one of these required security concerns. Several access control models are defined in the literature and this multitude of models creates heterogeneity of access control policies between the collaborating organizations. In this paper, we propose Access Control in Cross-Organizational coLLABoration ACCOLLAB, a solution for automatic mapping between heterogeneous access control policies in cross-organizational collaboration. To carry out this mapping, we suggest a mechanism founded mainly on XACML profiles and on a generic language derivative of XACML we define as Generic-XACML. We also formally prove that the mapping does not affect decision evaluation of policies. Thereby the proposed contribution ACCOLLAB allows each collaborating organization to communicate their access control policies and adopt other's policies without affecting their existing access control systems.

1. Introduction and Motivation

Collaborative activities have received a lot of attention from organizations due to the important need to address specific and common goals, to combine knowledge, skills, and experiences, to share resources (data, services, knowledge, and/or expertise) to meet a particular task. To succeed such collaboration, involved actors must first trust each other and communicate effectively to overcome the obstacles brought about by the benefits of collaboration.

During the last decade organizations have been more and more aware of the benefits of engaging in collaborative activities. Then in most of cases and in order to attain an ultimate objective or to answer required needs, they are obligated to share sensitive resources such as data, services, and knowledge. However, sharing sensitive and private resources, especially data and services, and exposing them for an external usage may prevent the organizations involved from collaborating. Hence, the focus on protecting data privacy and security issues in interorganizational collaboration represents a crucial requirement and becomes one of the most pressing concerns. Security issues aim at guaranteeing information availability, confidentiality, integrity, authenticity, and accountability. Data privacy known also as data protection aims to prevent sensitive information from being leaked or breached to unauthorized parties.

Several scientific research studies in the literature have raised this challenge, and identified that access control is one of the most important concerns of privacy and security. A number of access control models such as RBAC [1], TBAC [2], and ABAC [3] have been developed to address various aspects of access control problem.

In cross-organizational collaboration, additional requirements for access control arise like trust management, high level of privacy, interoperability, and dynamicity. Several access control solutions proposed in the literature have addressed this challenge. Some of them have proposed outright a new access control model [4, 5], or extended existing models to be suitable for cross-organizational collaboration [6, 7]. However, most of the suggested solutions require that

collaborating organizations profoundly modify their existing access control models, a situation that is difficult to achieve and is impractical in heterogeneous real systems. Other works have assumed that collaborating organizations are adopting the same access control model and proposed centralized solutions like [8, 9] or distributed solutions like [10, 11] to control access cross-organizations. Few works have tackled the heterogeneity of access control models problem [12–14], and none of them according to our knowledge has given a complete solution for automatic policy mapping between heterogeneous systems that covers both syntactic and semantic transformation.

Moreover, to enhance security interaction between organizations, we consider that enabling access control policy enforcement in customer organization is mandatory. However this property is not met by the evoked solutions. The need to enable provider policy enforcement by consumers system is motivated by multiple reasons:

(1) The need to ensure the fine grained access control defined by the provider policy. Usually a policy specifies fine grained constraints related to the subject who can access to a resource, but when the subject is in a foreign organization, the provider would be unable to determine the capability of the subject. Wherefore we need to enforce provider policy in the consumer side.

(2) The need to enforce context aware constraints defined by the provider policy when the policy specifies context constraints that could be determined only in the consumer organization.

(3) The need of high level of trustworthiness between collaborating organizations. Usually collaboration is regulated by contracts or agreement [15]. For a consumer organization, to keep a high level of trustworthiness, it should fulfill the provider policy, especially access control policy. To do so, consumer organization should be able to enforce the provider policy.

Reviewing the contributions presented in the literature in response to this challenge motivated us to believe in the need for a solution for collaborative access control that has the advantage to (1) tackle the heterogeneity in access control models, (2) allow automatic mapping of access control policies between collaborating organizations based on syntactic and semantic transformations, and (3) respect the legacy systems.

The aim of our contribution ACCOLLAB is to propose a new mechanism that ensures mapping between heterogeneous models automatically. This mechanism will help organizations to communicate their access control policies and adopt others' policies automatically without affecting the existing access control systems. In addition we have considered both syntactic and semantic mapping to propose a complete solution. To deal with semantic mapping, we have proposed an ontology-based semantic mapping process in [16]. In this paper, we focus on syntactic mapping, to which we have given a skeleton outline to syntactic mapping in a previous work [17].

The rest of this paper is organized as follows: Section 2 exposes related work, whereas Section 3 describes the mechanism of automatic mapping between access control models by means of XACML profiles and a proposed language Generic-XACML. In Section 4 we show in details how to map from XACML profiles to our Generic-XACML, while Section 5 is dedicated to present the reverse mapping. Finally, we conclude in Section 6.

2. Related Work on Access Control in Cross-Organizational Collaboration

In the literature, several contributions have addressed the problem of access control in cross-organizational collaboration. Some of them have proposed outright a new access control model, or extended existing models to be suitable for cross-organization collaboration. While the majority of works have assumed in their approaches that collaborating organizations are adopting the same access control model to propose architectures, frameworks, or solutions to control access cross-organizations, few works have tackled the heterogeneity of access control models problem, and none of them has given a complete automatic solution for policy mapping between heterogeneous systems. To more organize this section, we introduce as follows three cases that are as follows: Case 1: proposition of new access control model or extending an existing one; Case 2: solutions to control access across organizations adopting the same access control model; Case 3: approaches tackling the interoperability between heterogeneous models.

2.1. Case 1: Proposition of New Access Control Model or Extending an Existing One. Some works define of a new access control model or extend existing models in order to be suitable for cross-organization collaboration.

OrBAC [4] is an example of innovative models which is centered on the concept of Organization. Each access control policy is defined for and by an organization. OrBAC defines the notion of role, view, and activity that refer to subject, object, and action, respectively, from the perspective of an organization and includes also the notion of context. Using these concepts, policies are defined homogeneously in all collaborating organizations.

Authors in [18] propose a federated capability-based access control (FedCAC) system to tackle the challenges of access control for heterogeneous devices over IoT. They propose the delegation of domain-specific access control policies and identity management tasks from the centralized Policy Decision making Center PDC to fog computing nodes called coordinators. Authors in this work consider one homogeneous definition of access control policies and then they are synchronized among the PDC and coordinators.

Reference [6] is another example that extends RBAC model with new concepts required for collaborative environments in both intra- and interorganizations. Authors of that paper propose a generic access control ontology and a framework supporting administration and enforcement. The proposed model has been specified to protect data access in intra- and interorganizations collaboration, but it focuses on

organizations using only RBAC model and excludes other models.

Policies in these works will be defined in the same way for all collaborating organizations. Access requests will be homogeneous with enforcement mechanisms of the collaborating organizations. Meanwhile, adopting a new access control model requires rebuilding the whole access control system of collaborating organizations, which is impractical and sometimes refused by organizations.

2.2. Case 2: Solutions to Control Access across Organizations Adopting the Same Access Control Model. Many works have proposed solutions for access control in cross-organizational collaboration where all organizations adopt the same model (ABAC or RBAC are the most used). While reviewing the most interesting contributions we have concluded that two main architectures are proposed: centralized architecture and distributed architecture.

2.2.1. Centralized Architecture. The work [8] proposes a centralized architecture for access control across organizations where each collaborating organization defines policies associated to their shared resources. Then these policies are managed by a coordination organization depending on each collaboration incident and enforced by centralized components which bases on ABAC model.

Authors in [9] propose a Multiple-Policy supported Attribute-Based Access Control model (MPABAC) with a centralized architecture. This model extends the traditional ABAC model by providing cross-domain authentication and authorization. They propose a priority description to combine policies among multiple domains and adopt a hierarchical structure for policies enforcement.

Authors in [19] address the issues of combining multiple XACML policies in cross-organizational collaboration. They present a policy combination architecture that consists of classifying the rules based on attribute constraints in each policy of collaborative organizations and then reduce the rules of the corresponding classes to one with the same attribute constraints. The reduced rules are then combined into a new global policy by choosing the appropriate rule combining algorithm.

This kind of contributions proposed centralized solutions for access control in cross-organizational collaboration assuming that all collaborating organizations are using the same access control model. So they try to find a way to combine access control policies of collaborating organizations or to combine access control decisions.

2.2.2. Distributed Architecture. The work [10] proposes a policy distribution and synchronization schema for an IoT environment. It is based on virtual channels technique for the propagation and synchronization of policies across different domains in real-time. The paper presents a mechanism to dynamically enforce and propagate policies across heterogeneous domains. However it does not consider the heterogeneity of the policies themselves which can be expressed different ways according to each domain. It considers only ABAC model and assumes that no heterogeneity exists in policy definitions among different organizations.

Authors in [11] proposed a distributed access control architecture to address authorization issues across multiple clouds. The architecture is based on service-level agreement SLA component to allow peer to peer interoperation. SLA performs role mapping and evaluates policy constraints defined in a mediated SLA policy. This mediated policy is defined using RBAC XML-based declaration. Authors propose a solution for interoperability in multiple clouds collaboration assuming all clouds are adopting the same access control model RBAC.

Authors in [20] adapt and implement RBAC for a multidomain grid access control. Their approach includes an architecture for role mapping cross-domain based on role ranking mechanism. Authors consider only RBAC. Additionally this approach is not suitable for fine-grained authorization.

Authors in [21] address access control in dynamic cross-enterprise collaborations by proposing a framework for attribute and policy reconciliation, where attribute definitions or their interpretations are not standardized. The framework externalizes domain knowledge in order to dynamically infer attribute relationships during the evaluation of authorization decisions. Authors in this paper address the interoperability challenge for access control in cross-enterprise collaborations but they only consider ABAC model.

Even though these works give interesting solutions to manage access control in cross-organizations collaboration, they do not consider heterogeneity in access control models adopted by collaborating organizations.

2.3. Case 3: Approaches Tackling the Interoperability between Heterogeneous Models. An interesting work [22] proposed an ontological approach to deal with the interoperability between heterogeneous access control models by matching different ontologies that describe the diverse access control models of the interconnected organizations. Yet, authors focus on access control for cloud data storage when integrating heterogeneous organizations, which make it useless in a cross-organizational collaboration with segregated systems.

Authors in [12] address the heterogeneity problem of access control models across collaborating organizations. They proposed an equivalent based access collaboration model EABC to protect shared resources. This model covers multiple domains that are adopting different access control models and is based on defining equivalent access which involves entity mapping and entity linking relationships. They propose a formal definition of policy mapping across organizations. Unfortunately they do not give any details about mapping process.

Reference [13] proposed an enforcement architecture that evaluates the possibility of potential cross-domain policy deployment through model-driven mapping and translation using ontology-based mapping and query-based mapping. The paper presented a solution similar to ours. Meanwhile, it focuses on defined logical models, representing common operation rules, to ensure the semantic mapping. However, each logical model is defined by domain administrators,

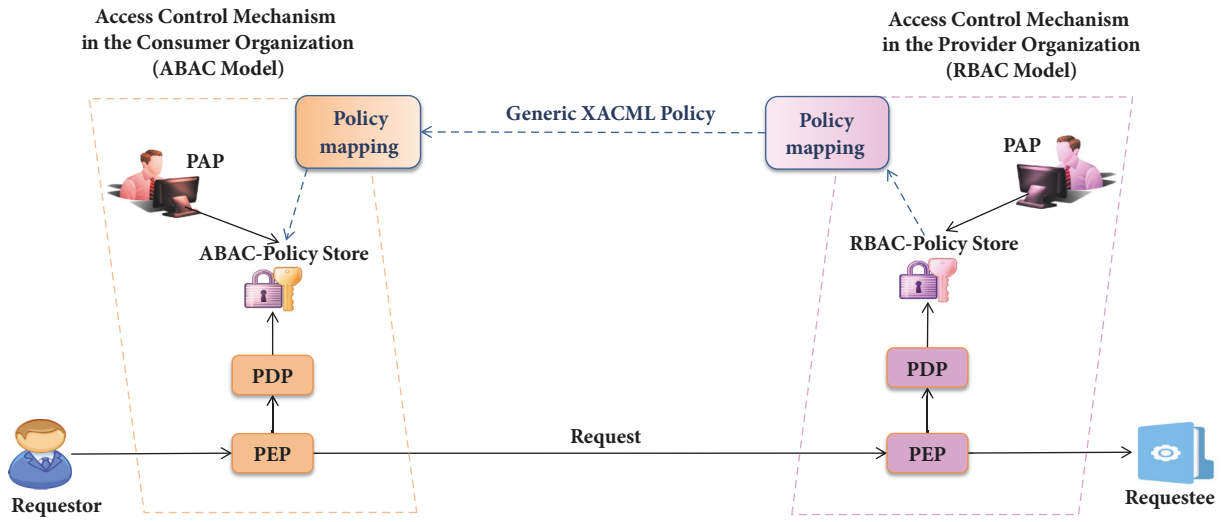

Figure 1: Architecture of policy mapping in cross-organizational collaboration.

which can generate heterogeneity in logical models themselves.

The paper [14] analyzed the common knowledge of access control models, and proposed an ontology-based model which can describe different access control models. This work gives a formal description of access control ontologies and proposes a connection algorithm, which is based on access ontology. However, neither details about the connection algorithm nor the mechanism of mapping between organizations' policies are provided, giving that each collaborating organization adopts its own access control mechanism.

These evoked contributions tackle the problem of access control in cross-organizational collaboration where each collaborating organization adopts a different access control model. Unfortunately none of them gives a complete solution using syntactic and semantic transformations.

This motivates us to come up with a solution characterized by

(1) Respect of legacy systems,

(2) Automatic policy mapping between collaborating organizations based on syntactic and semantic transformations,

(3) Tackling the heterogeneity in access control models.

3. Our Proposed Mechanism of Automatic Mapping between Heterogeneous Models

Our current contribution aims to suggest a solution for Access Control in Cross-Organizational coLLABoration (**ACCOLLAB**) that respects legacy systems of each organization in the collaboration and aims to enable the enforcement of providers' policies in the consumers' organizations. Figure 1 shows an example of two collaborating organizations using heterogeneous access control systems. The provider organization that offers a requestee (e.g., service, resource, data...) defines a policy using RBAC model and enforces access control using an adequate mechanism. So, the consumer organization that uses ABAC model and enforces access control using a different mechanism should be able to read provider's policy and enforces it using its own access control mechanism. Thus, we propose a mechanism for automatic policy mapping between organizations adopting heterogeneous access control models.

The automatic policy mapping involves two transformations: syntactic transformations that concern the form of the policy, which is our focus in this paper, and semantic correspondences we tackled in the previous contribution [16] Where we relied on a generic representation of access control concepts and proposed an ontology-based semantic mapping.

Thus, we assume, in this paper, that every single constraint in an access control policy expressed in an access control model has a semantic corresponding constraint in any other model and we focus on automatic mapping between models in term of policy definition.

To ensure an effective mapping we use XACML as an intermediate policy definition language for mapping. The motivation behind this choice is that XACML can be used to implement any access control model and that a number of XACML profiles are already defined.

Figure 2 depicts the global architecture of the mapping. Hence, to be able to map from a policy written according to a particular model to another model (e.g. RBAC model to ABAC model), we resort to XACML profiles as an intermediate language. So we define a high level syntax of XACML that we call **Generic-XACML** (detailed in Section 3.3). From this syntax we can switch to any XACML profile, and thereafter it will be translated to the target policy language which is specific to the model.

Our solution is distributed, but unlike existing distributed solutions [10, 11, 20, 21], we consider heterogeneous existing access control systems adopting heterogeneous models (ABAC, RBAC, UCON...). Our solution will be implemented as an additional layer on the top of existing access control

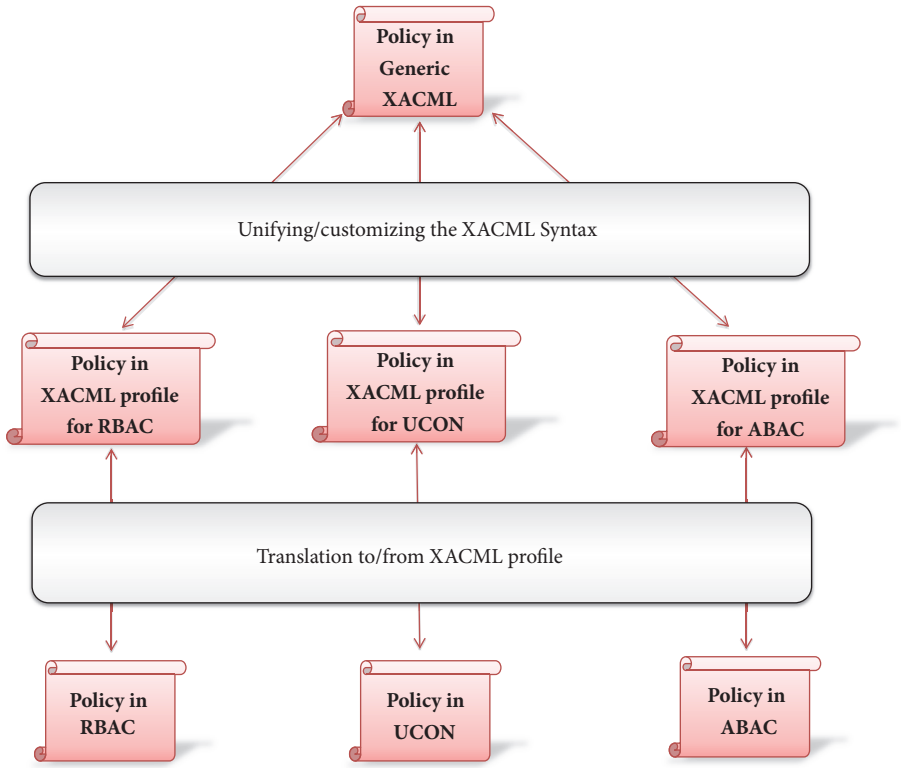

Figure 2: Mechanism of policy mapping between heterogeneous models.

systems; existing systems will not be changed only policies will be automatically translated.

In the next subsections we give an overview of XACML and XACML profiles. Then we give a definition of our generic-XACML language.

3.1. XACML: Overview. Recall that XACML (eXtensible Access Control Markup Language) [23] is a standardized access control policy and decision language based on XML. The core of XACML defines policies by hierarchical components. The root element is the PolicySet, it contains Policy or/and other PolicySet elements. Policy element contains a set of one or more Rule elements. A Rule element contains a condition that is evaluated to either True or False. A Rule element represents a single authorization or prohibition depending on its effect, which is either Permit or Deny. XACML provides **C**ombining **A**lgorithms that operate to combine decisions or effects of multiple Policy or Rule elements into a single decision via a Policy Combining Algorithm for Policy elements and via a Rule Combining Algorithm for Rule elements.

Rule, Policy and PolicySet elements include a Target element to specify their applicability to the access control request and optionally an obligationExpressions element or an adviceExpressions element to define obligations or advices respectively. The Target element may be empty or a conjunction of a disjunction (AnyOf elements) of a conjunction (AllOf element) of Subject, Resource, Action and/or Environment conditions expressed as Match elements. Subject, Resource, Action and Environment are the four attribute categories defined by XACML.

3.2. XACML Profiles

3.2.1. XACML-RBAC Profile. [24] defines a profile to meet the requirements for RBAC. The RBAC profile of XACML (XACML-RBAC) expresses a way to use the standard XACML within the RBAC model.

In this profile each Role is defined by a PolicySet element. It contains a Target element that makes the PolicySet applicable only to Subjects having the XACML Attribute associated with the given Role. The Target element does not restrict the Resource, Action, or Environment. This Role PolicySet element contains a unique PolicySet that defines the actual Permissionsassociated with the Role. Such a PolicySet contains PolicySet, Policy and Rule elements that describe the resources and actions that subjects are permitted to access, along with any further environmental conditions, such as time of day. A given Permission PolicySet may also contain references to Permission PolicySet elements associated with other Roles (hierarchy).

The Target element of a Permission PolicySet, and its included or referenced PolicySet, Policy, and Rule elements, must not limit the subjects to which the PolicySet is applicable.

3.2.2. XACML-UCON Profile. [25] defines a profile (XACML-UCON) for the use of XACML in expressing policies that

would ensure usage control as defined in UCON model. In this profile, Authorizations are specified by XACML Subject and XACML Resource in the Target element. Obligations are specified by XACML Condition. Conditions (the UCON concept) are specified by XACML Environment, Rights are specified by XACML Action. Continuity of usage decision will be expressed in the XACML Obligation within the Policy element. It would contain an AttributeAssignment which will specify the time interval between continuous policy re-evaluations.

Mutable Attributes are specified within XACML Obligations as XACML AttributeAssignment. The AttributeId is where the name of the mutable attribute is specified.

3.2.3. Other XACML Profiles. Other works like [26–28] define XACML profiles for Access Control List (ACL) and ABAC models. In the same way other profiles for other models can be developed, since XACML offers the possibility to express any concept as attributes. Thus, we can map any existing policy into the XACML policy language. The profile will specify the particularity of the model by specifying:

(i) The correlation between the model concepts and the categories of attributes,

(ii) The categories of attributes to put in for some Target elements,

(iii) The nesting of the XACML elements (specify the number of children of some elements).

(iv) The combining algorithms that are used.

3.3. Generic-XACML. When organizations engage in collaboration, access control policies related to the shared Requestees (services or resources...) are translated to the XACML profile for the model adopted by the provider organization. Then these policies are automatically mapped to Generic-XACML and shared jointly with the requestees. Later, these policies are automatically mapped to the XACML profile for the model adopted by the consumer organization and finally translated to the consumer model. So, Generic-XACML is a high level language that serves as intermediate for the mapping. Generic-XACML is inspired from XACML such as it matches the XACML specifications for policy definition and restricts the core XACML by the following constraints:

(i) It contains a root PolicySet element with an empty Target.

(ii) The root PolicySet contains exactly one nested Policy element with an empty Target as well.

(iii) The Policy element contains a set of nested Rule elements and optionally a set of Obligation and/or Advice elements.

Figure 3 depicts a pseudo code of the structure of a Generic-XACML policy.

In the next Sections 4 and 5, we show in more details how to map between Generic-XACML and XACML profiles. And we prove the equivalence between policies.

```
−<PolicySet>
    <!--any target-->
    <Target/>
  −<Policy>
      <!--any target-->
      <Target/>
    −<Rule>
        ...
        <!--optionnel-->
        <ObligationExpressions>...</ObligationExpressions>
        <!--optionnel-->
        <AdviceExpressions>...</AdviceExpressions>
      </Rule>
      <Rule>...</Rule>
      ...
    </Policy>
  </PolicySet>
```

Figure 3: A pseudo code of the structure of a Generic-XACML policy.

Table 1: Possible values of XACML elements.

	Match and Target value	Condition value	Rule, Policy and PolicySet value
$\top$	Match	True	Applicable (either permit or deny
$\bot$	Not match	False	Not applicable
I	Indeterminate	Indeterminate	Indeterminate

3.4. Policy Decision Evaluation for XACML and Generic-XACML. The Rule evaluation depends on the Target evaluation and the Condition evaluation [23]. The Target value can be either match, not match or indeterminate. The value indeterminate can be obtained if an error occurred or some required value was missing, so a decision cannot be made.

The Condition element is a set of propositional formulae which is evaluated to either True, False or Indeterminate. An empty Condition or an empty Target is always evaluated to True. The evaluation of a Rule element is either applicable, not applicable or Indeterminate. An applicable Rule has effect either deny or permit. Finally, the evaluation of Policy and PolicySet elements is based on a combining algorithm of which the result can be either applicable with its effect either deny or permit, not applicable or indeterminate.

In this paper, we refer to the formal XACML elements evaluation developed in [29]. In this work the authors use a three-valued logic represented by the three symbols: ($\top$, $\bot$, I) that correspond to XACML elements evaluation. Table 1 depicts the mapping between these three logic values and XACML elements evaluation.

In order to distinguish either an applicable policy permit access or deny it, this three-valued logic is extended to a multivalued logic represented by the set $V_6 = \{\bot, I_d, I_p, I_{dp}, \top_d, \top_p\}$, where the subscript d denotes Deny, the subscript p denotes Permit, and the subscript dp denotes Deny Permit.

Input: XACML profile document
Output: Generic-XACML document
Require: unified combining algorithm;
Create PolicySet element with empty Target;
Create Policy element with empty Target;
Parse the XACML document;
forall PolicySet element **do**
  forall Policy element **do**
    forall Rule element **do**
      Combine Rule Target with current policy and PolicySet Targets;
      Combine Rule obligationExpressions with current policy and PolicySet obligationExpressions;
      Combine Rule AdviceExpressions with current policy and PolicySet Targets AdviceExpressions;
      Insert current Rule in the Generic-XACML document;
Return Generic-XACML document;

Algorithm 1: Mapping from one XACML profile to Generic-XACML.

4. Mapping from XACML Profiles to the Generic-XACML

In this section we show that any policy written in an XACML profile can be mapped into our generic language. We explain how to proceed in order to map to the Generic-XACML without altering the logic of the policy and its decision evaluation. The following are steps of transformation of the original policy written in an XACML profile:

Step 1. Unifying the combining algorithms (in our study we focus on case where we have the **same combining algorithm** in all Policy and PolicySet elements).

Step 2. Nesting the Target of the Policy and PolicySet elements into their composite Rule elements and combining them with the Rule Target so that we obtain all Policy and PolicySet elements with an empty Target.

Step 3. Nesting of all ObligationExpression and AdviceExpression elements of the Policy and PolicySet elements into their composite Rule elements by inserting them into the ObligationExpressions element or into the AdviceExpressions element of the Rule.

Step 4. If a PolicySet is nested into another PolicySet, its Target is empty and its combining algorithm is the same as the container PolicySet; then it will be eliminated and substituted by its content.

Step 5. In order to obtain only one Policy element, we substitute all Policy elements by one Policy element that contains the content of all nested Rules together (they must have the same combining algorithm and an empty Target).

These steps can be carried out through Algorithm 1 that allows mapping from any XACML document to a Generic-XACML document. In the next subsections we prove that these transformations do not affect the decision evaluation of the policy.

Table 2: Rules truth table.

T	Ti	Ci	$T \wedge Ti$	Ri	Ri'
$\top$	$-$	$-$	Ti	$-$	Ri
I	$\top$ *or* I	$\top$ *or* I	I	$\top$ *or* I	I
I	$\bot$	$-$	$\bot$	$\bot$	$\bot$
I	$-$	$\bot$	$-$	$\bot$	$\bot$
$\bot$	$-$	$-$	$\bot$	$\bot$	$\bot$

4.1. Unifying the Combining Algorithms. To carry out the above transformations without affecting the global decision evaluation, we should have the same combining algorithm in the transformed elements. However, to come up with equivalence between combining algorithms, we need to extend XACML by proposing other elements. To avoid encumbering this paper we suppose we have the **same combining algorithm** in all Policy and PolicySet elements.

4.2. Policy and PolicySet Elements with an Empty Target. We prove that a Target of a Policy/PolicySet element can be nested to their composite Rule/Policy/PolicySet elements without changing the global decision evaluation. So that by repeating this transformation we obtain an empty Target for any Policy or PolicySet element.

Proof. Let $P = \langle T, R1 \dots Rn, \theta \rangle$ be a representation of a Policy where T is the Policy Target, $Ri = \langle Effect, Ti, Ci \rangle$ for $i \in [1 - n]$ are n nested Rules with Ti the Rule Target and Ci the condition for the Rule i, and θ is the combining algorithm.

And let $P' = \langle Null, R1' \dots Rn', \theta \rangle$ be the transformed Policy where the Target is empty and $Ri' = \langle Effect, T \wedge Ti, Ci \rangle$ for any $i \in [1 - n]$ are nested Rules with $T \wedge Ti$ is the conjunction of T and Ti.

We base on the truth tables (Tables 2 and 3) [23] to prove that the evaluation of the Policy P is the same as P': $[\boldsymbol{P}] = [\boldsymbol{P'}]$ we use the notation [] to express the evaluation of a Rule, Policy or PolicySet. □

Table 3: Policy truth table.

Target	*Rules*	*Policy*
$\top$	–	*Combining Algo*
$\boldsymbol{I}$	$\exists i \in [1-n]\ Ri = \top$ *or* I	I
$\boldsymbol{I}$	$\forall i \in [1-n]\ Ri = \bot$	$\bot$
$\bot$	–	$\bot$

Table 4: PolicySet truth table.

Target	*Policy or PolicySet*	*PolicySet*
$\top$	–	*Combining Algo*
$\boldsymbol{I}$	$\exists i \in [1-n]\ Pi = \top$ *or* I	I
$\boldsymbol{I}$	$\forall i \in [1-n]\ Pi = \bot$	$\bot$
$\bot$	–	$\bot$

Case 1. If $\boldsymbol{T} = \top$ then for any $i \in [1-n]$ $T \wedge Ti = Ti$ so the evaluation the nested Rules does not change $[Ri'] = [Ri]$ and then

$$\left[P'\right] = \left[\left\langle Null, R1' \ldots Rn', \theta\right\rangle\right] = \left[\left\langle \top, R1' \ldots Rn', \theta\right\rangle\right] \quad (1)$$

An empty Target always matches
Then

$$\left[P'\right] = [\langle \top, R1 \ldots Rn, \theta\rangle] = [P] \quad (2)$$

Case 2. If $\boldsymbol{T} = \boldsymbol{I}$ then

Case 2.1. If $\exists i \in [1-n]$ ($Ti = I$ Or $Ti = \top$) and ($Ci = \top$ or $Ci = I$) then $T \wedge Ti = I$
Then

$$\left[Ri'\right] = I \quad (3)$$

(because $Ci = \top$ or $Ci = I$)
So

$$\begin{aligned}\left[P'\right] &= \left[\left\langle Null, R1' \ldots Rn', \theta\right\rangle\right] \\ &= \left[\left\langle \top, R1' \ldots Rn', \theta\right\rangle\right] \\ &= comAlg\left(\left[R1'\right] \ldots \left[Rn'\right]\right)\end{aligned} \quad (4)$$

(*comAlg* is the function that evaluates decisions of $[R1'] \ldots [Rn']$ according to the combining algorithm used)
So

$$\left[P'\right] = I \quad (5)$$

(at least one Rule evaluated to Indeterminate)
On the other hand $[Ri] = \top$ or I then $[P] = I$ (Target = I)
So

$$\left[P'\right] = [P] \quad (6)$$

Case *2.2.* If for any $i \in [1-n]$ ($Ti = \bot$ O r $Ci = \bot$) then for any $i \in [1-n]$ $[Ri'] = [Ri] = \bot$

$$\begin{aligned}\left[P'\right] &= \left[\left\langle Null, R1' \ldots Rn', \theta\right\rangle\right] \\ &= \left[\left\langle \top, R1' \ldots Rn', \theta\right\rangle\right] = \bot\end{aligned} \quad (7)$$

And

$$[P] = [\langle I, R1 \ldots Rn, \theta\rangle] = \bot = \left[P'\right] \quad (8)$$

Case 3. If $\boldsymbol{T} = \bot$ then for any $i \in [1-n]$ $T \wedge Ti = \bot$
So for any $i \in [1-n]$ $[Ri'] = \bot$ and $[Ri] = \bot$
Then

$$\begin{aligned}\left[P'\right] &= \left[\left\langle Null, R1' \ldots Rn', \theta\right\rangle\right] \\ &= \left[\left\langle \top, R1' \ldots Rn', \theta\right\rangle\right] \\ &= comAlg\left(\left[R1'\right] \ldots \left[Rn'\right]\right) = \bot\end{aligned} \quad (9)$$

And

$$[P] = [\langle \bot, R1 \ldots Rn, \theta\rangle] = \bot = \left[P'\right] \quad (10)$$

The same reasoning for a PolicySet composed by a set of policies or PolicySets with the truth Table 4.

4.3. Policy and PolicySet Elements with No Obligation or Advice Elements. Obligation or Advice are operations that must be fulfilled in conjunction with an authorization decision (permit or deny authorization decision). ObligationExpression or AdviceExpression elements may be added optionally in a Rule, Policy, or PolicySet elements.

Obligation and Advice do not affect the access decision but they are fulfilled when the access decision is equal to the value specified in the FulfillOn attribute for Obligation element and AppliesTo attribute for Advice element.

So since Obligation and Advice do not affect the access decision we can imbricate them into the nested Rule elements. This results a redundancy in ObligationExpression and AdviceExpression elements but it will be overcome when mapping to another XACML profile.

4.4. Substitute Nested PolicySet Elements by Their Contents. Generic-XACML is based on XACML but defines a specific arborescence of the elements. It contains a root PolicySet with an empty Target and a nested Policy element that has an empty Target as well and a set of nested Rule elements. In Section 4.2 we have proved that a Target of a Policy/PolicySet can be nested to their composite Rules/Policies/PolicySets without changing the global decision evaluation. In this section we prove that if a PolicySet is nested into another PolicySet, its Target is empty, and its combining algorithm is the same as the container PolicySet then it can be eliminated and substituted by its contents as illustrated in Figure 4.

Proof. Let *CPS* be the Container PolicySet element and *NPSi* with $i \in [1-n]$ be its Nested PolicySet elements. All of container and nested PolicySet elements have an empty Target.

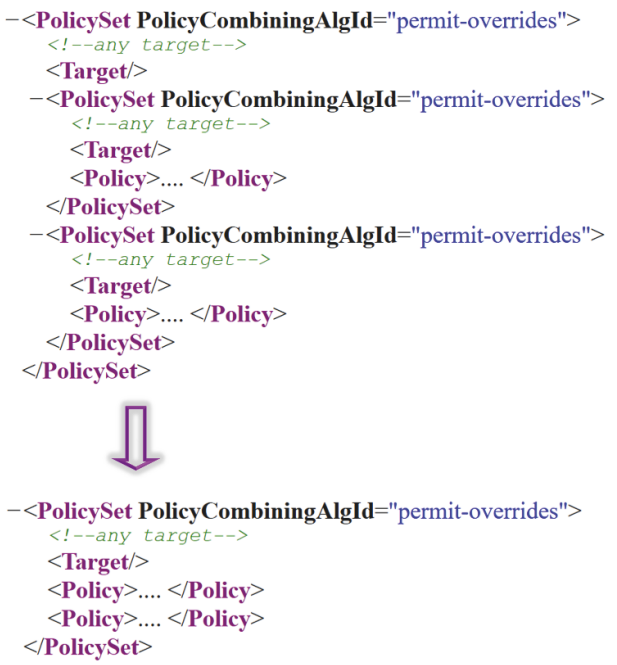

Figure 4: Substitute nested PolicySet elements by their contents.

comAlg is the function that evaluates a set of decisions according to the combining algorithm used. Then the decision evaluation of the *CPS* is

$$[CPS] = comAlg([NPS1], \ldots, [NPSn]) \tag{11}$$

And

$$[NPSi] = comAlg([Pi1], \ldots, [Pim_i]) \quad for\ any\ i \in [1-n] \tag{12}$$

$Pi1 \ldots Pim_i$ are nested policies for the PolicySet *NPSi* and m_i is their number.

Let us prove that

$$[CPS] = comAlg([P11], \ldots [P1m_1], \ldots [Pn1], \ldots [Pnm_n]) \tag{13}$$

We use the multivalued approach presented in [29] where they define for each combining algorithm a lattice $(V_6, \leq_{CA})$, where V_6 is the set $\{\bot, I_d, I_p, I_{dp}, \top_d, \top_p\}$ and the ordering $\leq_{CA}$ is defined according to the combining algorithm specification.

So the combining algorithm function applied to a set S of V_6 is the least upper bound: the supremum (*sup*) of S.

Then

$$\begin{aligned}[CPS] &= sup([NPS1], \ldots [NPSn]) \\ &= sup(sup([P11], \ldots [P1m_1]), \\ &\ldots sup([Pn1], \ldots [Pnm_n]))\end{aligned} \tag{14}$$

If we have the same combining algorithm, the same ordering for every *NPSi*, then

$$\begin{aligned}&sup(sup([P11], \ldots [P1m_1]), \\ &\ldots sup([Pn1], \ldots [Pnm_n])) = sup([P11], \\ &\ldots [P1m_1], \ldots [Pn1], \ldots [Pnm_n])\end{aligned} \tag{15}$$

Then

$$\begin{aligned}&[CPS] \\ &= sup([P11], \ldots [P1m_1], \ldots [Pn1], \ldots [Pnm_n])\end{aligned} \tag{16}$$

□

So if we eliminate all nested PolicySet elements and substitute them by their nested Policy elements the decision evaluation does not change.

4.5. Merging All Policy Elements into One Policy. Now we prove that all nested Rules can be merged into only one Policy element if all Policy elements have the same combining algorithm. We show that this transformation does not affect the decision evaluation of the container PolicySet.

Proof. Let *CPS* be the Container PolicySet and *NPi* for $i \in [1-n]$ be the nested Policy elements and *NP* the resulting nested Policy. All of these elements, container PolicySet, the nested policies and the resulting Policy, have an empty Target and the same combining algorithm.

The evaluations of *CPS*, *NPi*, and *NP* are

$$[CPS] = comAlg([NP1], \ldots [NPn]) \tag{17}$$

$[NPi] = comAlg([Ri1], \ldots [Rim_i])$ for any $i \in [1-n]$, where $Ri1 \ldots Rim_i$ are nested Rules for the Policy *NPi*

$$\begin{aligned}[NP] &= comAlg([R11], \ldots [R1m_1], \ldots [Rn1], \\ &\ldots [Rnm_n])\end{aligned} \tag{18}$$

We prove that

$$[CPS] = [NP] \tag{19}$$

If we follow the same reasoning as above and we suppose we have the same combining algorithm for all policies: the same ordering, we can prove that

$$\begin{aligned}[CPS] &= sup([NP1], \ldots [NPn]) \\ &= sup(sup([R11], \ldots [R1m_1]), \\ &\ldots sup([Rn1], \ldots [Rnm_n])) = sup([R11], \\ &\ldots [R1m_1], \ldots [Rn1], \ldots [Rnm_n]) = [NP]\end{aligned} \tag{20}$$

□

5. Mapping from Generic-XACML to XACML Profiles

Once we obtain the Generic-XACML policy we can reform it into the desirable XACML profile. This involves encompassing Rules into policies and PolicySets according to the profile

```
Input: Generic-XACML document
Output: XACML-RBAC document
Create a root PolicySet in XACML-RBAC document with an empty Target
For i=1 to rulesnumber do
   Parse Target of rule i in Generic-XACML document;
      If Target designate the Subject then
         currentValue:=value(Subject);
         Append a role PolicySet with Target designating CurrentValue for the Subject;
         Insert a Permissions PolicySet with an empty Target;
         Insert a policy with an empty Target;
         RoleRules[]:= rule i;
         For j=i+1 to rulesnumber do
            Parse Target of rule j;
            If value(Subject)= currentValue then
               RoleRules[]:= rule j;
         Alter RoleRules Targets; //delete constraint about currentValue;
         Insert RoleRules into the policy;
Return XACML-RBAC document;
```

Algorithm 2: Mapping from Generic-XACML to XACML-RBAC profile.

specifications. In this section, we describe how to map from Generic-XACML to a specific XACML profile. We follow two great steps:

(1) Reproducing a customized policy conform to the profile specifications.

(2) Optimizing the resulting policy.

For both steps, the sorts of transformations we carry out are as follows:

(i) Inserting container Policy or PolicySet elements having an empty Target element and the same combining algorithm as the initial policy.

(ii) Moving constraints that are common between the nested elements from their Targets to the Target of the container element.

(iii) Moving the ObligationExpression or AdviceExpression elements that are common between the nested elements to the container element.

These transformations do not affect the decision evaluation of the global policy as it is proved in the Section 4.

5.1. Conformance to Specific Profile. For a generic policy to be conforming to profile specifications, it is transformed and customized by a specific algorithm that depends on the profile, and that differs from one profile to another. For illustration purposes we touch on RBAC and UCON profiles.

5.1.1. Mapping from Generic-XACML to XACML-RBAC Profile. To translate a policy from Generic-XACML into XACML-RBAC profile we follow Algorithm 2.

In conformance with XACML-RBAC profile specifications, the resulting document will contain a root PolicySet element with an empty Target. On the other side, the original document is parsed. Then Targets of the Rules are browsed. So, for each possible value of the Subject we create a PolicySet element representing a Role and a nested PolicySet element with an empty Target representing the Role permissions. As for the PolicySet representing the Role, the Target will contain an imbrication of an AnyOf, an AllOf, and a Match element. The latter designates the current value of the Subject. Then, all Rules that contain a Match element that satisfy the current Subject are selected and inserted in a Policy element nested into the PolicySet representing permissions. Before rules are inserted, their Targets are altered in such a way to eliminate the Match element that designates the current value of the Subject.

Rules containing no Match element that designates a Subject are considered as Rules concerning all Subjects. Then these Rules are inserted into every PolicySet elements representing a role.

Special Case. Algorithm 2 consists of factoring Rules into policies. The factor is a Match element that designates the Subject. This is only possible if Targets are logically expressed as conjunction of Match elements. If an AnyOf element contains more than one AllOf element the Target will be evaluated as a conjunction of disjunction of conjunction of Match elements. We cannot factorize by Match element. In this case, our algorithm will compare the AnyOf element as a whole rather than comparing only the Match element.

5.1.2. Mapping from Generic-XACML to XACML-UCON Profile. XACML-UCON profile described in [25] is an implementation for UCON_{ABC} model [30] that fulfills XACML specifications and categorizes policies into 3 types: Authorizations, oBligations, and Conditions. The particularity of UCON is the **C**ontinuity of **U**sage and **M**utable Attributes that are expressed as XACML Obligations within the Policy element. So, mapping to UCON profile consists of categorizing original Rules into three categories: Authorizations A, oBligations B, and Conditions C. Then each category is

```
Input: Generic-XACML document
Output: XACML- UCON document
create a root PolicySet in XACML-UCON document
for i=1 to rulesnumber do
   parse Target of rule i in Generic-XACML document;
      if ∃ any element designating Environment attribute then
         if ∃ obligation specifying the request interval for ongoing control then
            onCRules[]:=rule i;
         else
            preCRules[]:=rule i;
      else if ∃ condition element in rule i then
         if ∃ obligation specifying the request interval for ongoing control then
            onBRules[]:=rule i;
         else
            preBRules[]:=rule i;
      else
         if ∃ obligation specifying the request interval for ongoing control then
            onARules[]:=rule i;
         else
            preARules[]:=rule i;
if preARules is not empty then
   insert preAPolicy;
   insert preARules into preAPolicy;
if onARules is not empty then
   insert onAPolicy;
   insert onARules into onAPolicy;
if preBRules is not empty then
   insert preBPolicy;
   insert preBRules into preBPolicy;
if onBRules is not empty then
   insert onBPolicy;
   insert onBRules into onBPolicy;
if preCRules is not empty then
   insert preCPolicy;
   insert preCRules into preCPolicy;
if onCRules is not empty then
   insert onCPolicy;
   insert onCRules into onCPolicy;
return XACML-UCON document;
```

Algorithm 3: Mapping from Generic-XACML to XACML-UCON profile.

divided into two subcategories pre- (evaluated only once) and ongoing (Continuous re-evaluation). So the resulting document will contain eventually six types of Policy: preA, preB, preC, onA, onB, and onC.

The Algorithm 3 shows how to execute the mapping from Generic-XACML to XACML-UCON profile. Rules of the original document are parsed. So, if a Rule contains at least one element that designates or selects an XACML Environment attribute, it will be considered as a Condition C. If it contains an XACML Condition element, it will be considered as an oBligation B. Otherwise it will be considered as an Authorization A. On the other hand, if the Rule contains an XACML Obligation element with an AttributeAssignment element which specifies the time interval between continuous policy re-evaluations, the Rule is inserted in the resulting document inside an ongoing Policy onA, onB, or onC. Otherwise it is inserted inside a pre Policy preA, preB, or preC.

5.2. Optimizing Policies. When mapping from Generic-XACML to any XACML profile, the resulting policy could have redundancies in Target, Advice, or Obligation elements, or a large number of Rules in one Policy element, which makes it costly for the Policy Decision Point in term of time execution. In this subsection, we propose an algorithm to optimize the resulting policies.

This algorithm (Algorithm 4) consists of regrouping Rules into policies based on common values of attributes in the Target element. Therefore, the algorithm iterates over attribute categories. The attribute categories and their order are selected based on the model (e.g., we follow the order: Resource, Action, then Environment for RBAC model). So, for each attribute category and for each Policy element, Rules are parsed. Then, if the Rule Target designates current attribute category, its value is compared with attribute category value of other Rules. Thus, Rules with equal attribute

```
Input: XACML document
Forall attribute categories do
  Forall Policy elements do
    evaluate rulesnumber of current policy;
    If rulesnumber ≥ 2 then //policy with one rule does not need optimization
      For i=1 to rulesnumber do
        parse Target of rule i;
        If Target designates current attribute category then
          CurrentValue:=value(attribute category);
          combinedRules[]:= rule i;
          For j=i+1 to rulesnumber do
            parse Target of rule j;
            If value(attribute category)= CurrentValue then
              combinedRules[]:= rule j;
          If length(combinedRules) ≥ 2 then
            If length(combinedRules) = rulesnumber then
              alter Target of current Policy element;
              alter combinedRules Targets;
            Else
              create sibling policy with Target designating CurrentValue for attr category;
              alter combinedRules Targets;
              move combinedRules to the new sibling policy;
```

Algorithm 4: Optimizing policies.

category values are combined into a sibling Policy (new Policy having the same parent as the current Policy). Its Target will contain an imbrication of an AnyOf, an AllOf, and a Match element. The latter designates the current value of the current category of attribute.

If a resulting document contains a large number of Policy elements, this algorithm can be extended to combine Policy elements into PolicySet elements with the same instructions.

Other constraints can be added to the algorithm depending on the profile specifications (e.g., Target of PolicySet representing a role in RBAC profile designates only Subject attributes).

Special Case. Similarly to Algorithm 2, if an AnyOf element contains more than one AllOf element, the algorithm will compare the AnyOf element as a whole rather than comparing only the Match element.

As for obligations and advices, if all Rules within a Policy element have the same ObligationExpression or the same AdviceExpression element then this expression is moved to the parent Policy.

6. Conclusion and Future Research

In this paper, we propose Access Control in Cross-Organizational coLLABoration (ACCOLLAB) that tackles the problem of heterogeneity of access control models cross-organizations while respecting the internal access control model of each involved organization in a collaboration.

This solution is based on a mechanism for automatically mapping between policies in different models. We considered the previously proposed ontology-based semantic mapping process to deal with semantic correspondences and focus on syntactic transformations of the heterogeneous policies to propose a complete solution.

This automatic mapping is based on XACML profiles and the generic language Generic-XACML we have defined. Thus, we have given a logic proof for all of the mapping steps.

Thus, our generic access control model ACCOLLAB solves the heterogeneity problem, ensures the interoperability cross-organizations, and maintains the privacy of each collaborating organization.

We are working to implement our mapping algorithms using the XACML implementation Balana; then we are going to complete the implementation of the policy mapping architecture based on WSO2 servers.

Besides that, we intend to extend XACML in order to find equivalence between combining algorithms to make our proposed mechanism covering heterogeneous combining algorithms.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *The Computer Journal*, vol. 29, no. 2, pp. 38–47, 1996.

[2] R. K. Thomas and R. S. Sandhu, "Task-based authorization controls (TBAC): a family of models for active and enterprise-oriented authorization management," in *Database Security XI*, IFIP Advances in Information and Communication Technology, pp. 166–181, Springer US, Boston, MA, 1998.

[3] E. Yuan and J. Tong, "Attributed based access control (ABAC) for web services," in *Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, pp. 561–569, IEEE, July 2005.

[4] A. A. E. Kalam, R. E. Baida, P. Balbiani et al., "Organization based access control," in *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, pp. 120–131, 2003.

[5] A. Kalam and Y. El, "Multi-OrBAC: a New Access Control Model for Distributed, Heterogeneous and Collaborative Systems," in *Proceedings of the 8th IEEE International Symposium on Systems and Information Security*, p. 1, 2006.

[6] A. Kamoun and S. Tazi, "A semantic role-based access control for intra and inter-organization collaboration," in *Proceedings of the 23rd IEEE International WETICE Conference, WETICE 2014*, pp. 86–91, Italy, June 2014.

[7] W. Zhou and C. Meinel, "Team and task based RBAC access control model," in *Proceedings of the 2007 Latin American Network Operations and Management Symposium - LANOMS 2007*, pp. 84–94, Brazil, September 2007.

[8] J. Li, J. Zic, N. Oakes, D. Liu, and C. Wang, "Design and evaluation of an integrated collaboration platform for secure information sharing," in *Cooperative Design, Visualization, and Engineering*, vol. 9929 of *Lecture Notes in Computer Science*, pp. 185–193, Springer International Publishing, Cham, 2016.

[9] F. Liang, H. Guo, S. Yi, and S. Ma, "A multiple-policy supported attribute-based access control architecture within large-scale device collaboration systems," *Journal of Networks*, vol. 7, no. 3, pp. 524–531, 2012.

[10] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, "Dynamic policies in internet of things: enforcement and synchronization," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2228–2238, 2017.

[11] A. Almutairi, M. Sarfraz, S. Basalamah, W. Aref, and A. Ghafoor, "A distributed access control architecture for cloud computing," *IEEE Software*, vol. 29, no. 2, pp. 36–44, 2012.

[12] H. Xiang, X. Xia, H. Hu, J. Sang, and C. Ye, "Approaches to access control policy comparison and the inter-domain role mapping problem," *Information Technology and Control*, vol. 45, no. 3, pp. 278–288, 2016.

[13] Z. Wu and L. Wang, "An innovative simulation environment for cross-domain policy enforcement," *Simulation Modelling Practice and Theory*, vol. 19, no. 7, pp. 1558–1583, 2011.

[14] Z.-W. Wang, "A generic access control model based on ontology," in *Proceedings of the 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, WCNIS 2010*, pp. 335–339, China, June 2010.

[15] S. Haguouche and Z. Jarir, "Managing heterogeneous access control models cross-organization," in *Proceedings of the International Conference on Risks and Security of Internet and Systems*, pp. 222–229, 2015.

[16] S. Haguouche and Z. Jarir, "Generic access control model and semantic mapping between heterogeneous policies," *International Journal of Technology Diffusion (IJTD)*, vol. 9, no. 4, pp. 52–65, 2018.

[17] S. Haguouche and Z. Jarir, "Toward a generic access control model," in *Proceedings of the 3rd IEEE World Conference on Complex Systems, WCCS 2015*, pp. 1–6, IEEE, Morocco, November 2015.

[18] R. Xu, Y. Chen, E. Blasch, and G. Chen, "A federated capability-based access control mechanism for Internet of Things (IoTs)," in *Proceedings of the Sensors and Systems for Space Applications XI. International Society for Optics and Photonics*, 2018.

[19] L. Duan, Y. Zhang, S. Chen et al., "Automated policy combination for secure data sharing in cross-organizational collaborations," *IEEE Access*, vol. 4, pp. 3454–3468, 2016.

[20] G. Geethakumari, A. Negi, and V. N. Sastry, "A cross - Domain role mapping and authorization framework for RBAC in grid systems," *International Journal of Computer Science and Application*, vol. 6, no. 1, pp. 1–12, 2009.

[21] D. Preuveneers, W. Joosen, and E. Ilie-Zudor, "Policy reconciliation for access control in dynamic cross-enterprise collaborations," *Enterprise Information Systems*, vol. 12, no. 3, pp. 279–299, 2018.

[22] C. Esposito, "Interoperable, dynamic and privacy-preserving access control for cloud data storage when integrating heterogeneous organizations," *Journal of Network and Computer Applications*, vol. 108, pp. 124–136, 2018.

[23] OASIS XACML Technical Committee, "eXtensible Access Control Markup Language (XACML) Version 3.0," http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html, accessed July 2018.

[24] OASIS XACML Technical Committee, "XACML v3. 0 core and hierarchical Role Based Access Control (RBAC) profile version 1.0 (committe specification 02)," http://docs.oasis-open.org/xacml/3.0/rbac/v1.0/xacml-3.0-rbac-v1.0.html, accessed July 2018.

[25] Y. Ghazi, R. Masood, M. A. Shibli, and S. Khurshid, "Usage-based access control for cloud applications," in *Innovative Solutions for Access Control Management*, pp. 197–223, 2016.

[26] G. Karjoth, A. Schade, and E. Van Herreweghen, "Implementing ACL-based policies in XACML," in *Proceedings of the 24th Annual Computer Security Applications Conference, ACSAC 2008*, pp. 183–192, USA, December 2008.

[27] M. Xu, D. Wijesekera, and X. Zhang, "Towards session-aware RBAC administration and enforcement with XACML," in *Proceedings of the 2009 IEEE International Symposium on Policies for Distributed Systems and Networks*, 2009.

[28] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering DAC, MAC and RBAC," *Lecture Notes in Computer Science*, vol. 7371, pp. 41–55, 2012.

[29] C. D. P. K. Ramli, H. R. Nielson, and F. Nielson, "The logic of XACML," *Science of Computer Programming*, vol. 83, pp. 80–105, 2014.

[30] J. Park and R. Sandhu, "The UCON ABC usage control model," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 128–174, 2004.