Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MODALITY CONFLICT DETECTION MODEL WITH AUTHORIZATION PROPAGATION IN POLICY EVALUATION

TEO POH KUANG[1], HAMIDAH IBRAHIM[2], FATIMAH SIDI[2], AND NUR IZURA UDZIR[2]
Department of Computer Science, Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
E-mail: [1]pohkuang1985@yahoo.com.my, [2]{hamidah.ibrahim; fatimah; izura}@upm.edu.my

ABSTRACT

Modality conflict is one of the main issues in policy evaluation. Existing modality conflict detection approaches do not consider complex condition attributes such as spatial and temporal constraints. In this paper, a modality conflict detection model is proposed to identify the applicable policies during policy evaluation, which supports an authorization propagation rule to investigate the class-subclass relationships of a subject, resource, action, and location of a request and a policy. We have evaluated the effectiveness of our proposed modality conflict detection model on real XACML policies for university, conference management, and health-care domain. Overall, our solution achieved higher percentage of *P*, *R,* and *F* in retrieving the applicable policies and in detecting modality conflict as compared to the previous work.

Keywords: *Modality Conflict*, *Authorization Propagation*, *Policy Evaluation*, *Spatial and Temporal Constraints*, *Distributed Environment*

1. INTRODUCTION

Policy evaluation is a process to determine whether a request satisfies the access control policies. A policy is said to be applicable to a request if the attribute values of the request are matched with the attribute values of the policy; which defined what should, and what should not, be allowed, and, in some sense, to different definitions of what ensuring security means [35]. Due to the dynamism and complexity of collaborative applications, the authoring and implementation of policies are usually a distributed process [4] since each of the distributed organization would likely be designing their policies autonomously to serve their particular authority principle concern regardless of which parties have already joined the coalition; which unlikely require an interoperability process to enable mutual understanding among parties. Because of the complexity of semantic policy repositories and the variation of user access privileges, conflict may arise in a group of access control policies.

With the increasing popularity of distributed systems and collaborative applications, there is a need to apply a conflict analysis method in policy evaluation. Modality conflict is an issue in policy evaluation which arises because of the existence of both positive and negative authorizations for a given subject-object[1] pair in policy evaluation. Past works [7, 9, 10, 23, 25] rely on the conventional modality conflict detection process that does not exploit the semantic relationships among the policy attributes, i.e. subject, action, resource, and condition. Obviously, authorizations can be propagated based on the hierarchical structures of policy attributes that permit inheritance relationships between concepts. Exploring these semantic relationships to ensure consistency in authorization decision is crucial as multiple inheritance paths in the hierarchy may lead to the same requested attribute value with different authorization decisions [26].

Generally, in a large distributed system, when a user sends a request to execute an action, if there is no explicit authorization specified for the user, there must be some way to propagate authorizations for the user [17]. Several works have been devoted to the topic of propagation of authorizations in distributed

[1] The terms object and resource are being used interchangeably in this paper.

systems according to the inheritance relationships between concepts [1, 5, 8, 9, 17, 26, 36]. However, the concern of these works is only on the authorization propagation on the subject, resource, or action attributes. Complex condition elements such as semantic relationships between spatial or temporal elements are necessary to take into account in the modality conflict detection process.

In this paper, we focus on the eXtensible Access Control Mark-up Language (XACML) which is based on Discretionary Access Control (DAC) model since it is highly flexible and currently is the most widely used [18]. This paper is a continuation and refinement of our earlier preliminary work [20]. We propose a modality conflict detection model to identify the applicable policies, which relies on inheritance relationships between the attribute values of a request and a policy. The modality conflict detection model contains subject, resource, action, location hierarchies that supports a more adequate representation of their semantics. We mainly focus on a process before the actual policy evaluation is performed to filter out the irrelevant policies which assists the policy administrators to resolve modality conflict among these potentially applicable policies according to their priority to better protect sensitive and private data.

Overall, the main contributions of this work are briefly described as follows:

- An *applicable policies module* that aims at identifying all possible policies which include both the explicit and implicit policies for a given request submitted by a user.
- A *modality conflict module* that aims at detecting and resolving modality conflict among the applicable policies.
- Both modules are embedded into a *modality conflict detection model* which aims to effectively achieve accurate authorization decision.
- Extensive experiments are conducted and the experimental results of the proposed solution are presented to prove its capability of retrieving the applicable policies and detecting modality conflict during policy evaluation and eventually achieve accurate authorization decision.

The rest of this paper is organized as follows. Section 2 reviewed the authorization propagation rules that are proposed by previous studies for detecting modality conflict. Section 3 presents in details the proposed modality conflict detection model. Section 4 presents the evaluation of the proposed solution by evaluating the performance of the modality conflict detection model and the results are compared to the previous work. Section 5 concludes the current study and sheds light on some directions which can be followed in the future.

2. RELATED WORK

An access control is a mechanism to secure a resource (e.g. information) in a system by restricting the operations a subject is allowed to perform on the resource [33]. To gain access to a resource, an access control policy is enforced which determines the conditions to be fulfilled by a subject, i.e. only authorized subjects that are assigned permissions can perform actions on that resource [32]. Based on the hierarchy derivation policies [11], implicit policies can be derived through authorization propagation. However, this creates the possibility to retrieve multiple applicable policies with different effects, positive (permit) or negative (denial) authorizations, to an authorization decision which may lead to modality conflict. Although authorization propagation is a convenient and easy way to specify implicit policies, it can result in unforeseen conflicts that need to be detected and resolved [19].

There are several works like [3, 13, 14, 22, 29, 30, 31] that fully rely on the conventional modality conflict detection which detects conflicts only among the existing explicit policies; which caused incomplete modality conflict detection. While works like [1, 8, 16, 19, 28, 34, 36, 38, 39, 41] focus mainly on policy conflict detection and resolution among a set of policies in a policy database once a new party joined the collaboration that can be used by policy administrators to proactively detect conflict policies.

The issues related to authorization propagation based on inheritance relationships between concepts have been explored by [1, 5, 8, 9, 17, 26, 36]. Jajodia, S. *et al.* [17] exploit the hierarchical structures of attributes (roles, user group, and resources) and four types of derivation rule to propagate the authorizations of a node to all its descendants in the hierarchy; which allows the specification of both positive and negative authorizations. The notions of authorization derivation, conflict resolution, and authorization decision strategies are incorporated into their proposed unified framework.

In Damiani, E. *et al.* [9], the notion of domain nesting principle, which is simply the hierarchy concept used for propagating authorizations, is applied to resolve modality conflict that arises from the "part-of" relations. Meanwhile, Bertino, E. *et a1.*

[5] employ the hierarchy of an organization to propagate the access permissions; where the access permissions given to a role in a higher position subsumed the access permissions given to all roles with a lower position. Through the authorization propagation concept, a requested node may inherit permissions of opposite sign from its parent nodes, in which implicit permissions are derived which may cause inconsistencies. Mohan, A. *et al*. [26] applied descending propagation for cases where the child nodes have authorization decision different from the requested resource node. If the response is "Deny" for any of the parent requested resource nodes, then the authorization decision is returned as "Deny". In Brodecki, B. *et al*. [8], with the knowledge of the hierarchy of subjects and resources, an algorithm is proposed to discover modality conflict among policies with opposite authorization decisions when descending propagation is applied. Shaikh, R. A. *et al*. [36] proposed a novel method which applies the concept of role hierarchy and permission inheritance to detect modality conflict among the access control policies.

The authorization propagation presented in [1, 5, 8, 9, 17, 26, 36] is limited to the subject, resource, and action attributes, thus affects the result of authorization decision since modality conflict is not completely detected. Moreover, these works are limited to simple condition evaluation in which string equal function is used.

Adi, K. *et al*. [1] argued that policy permission should not only depend on the subject, resource, and action inheritance relationships, but also restricted by considering the context information which include the spatial and temporal constraints. Hence, to ensure adequate protection of services and resources, the context information such as the requestor's location information (spatial constraints) and the requestor's access time (temporal constraints) are necessary to be considered in an access control policy as well as the modality conflict detection process [2].

3. THE MODALITY CONFLICT DETECTION MODEL

In this section, we present our proposed *modality conflict detection model* which consists of two main module, namely: (i) *applicable policies module* that identifies the applicable policies which include both the explicit and implicit policies for a given request submitted by a user, and (ii) *modality conflict module* that detects modality conflict among the applicable policies. Figure 1 shows the overall general process flow of the proposed *modality conflict detection model*, which is further elaborated in the following subsections, while Table 1 and Table 2 present the university policies and requests, respectively that are used as an illustrative example.

3.1 Identifying the Applicable Policies

In ensuring the security of a large authorization system, multiple sets of policies, $PS = \{PS_1, PS_2, \ldots, PS_n\}$, each consisting of multiple policies, $PS_i = \{P_1, P_2, \ldots, P_m\}$, with thousands of rules, $RP_i = \{R_1, R_2, \ldots, R_k\}$ may be specified by a number of authorities. Given a request, *Req*, submitted by a user, $Subject_{req}$, with the intention to access, $Action_{req}$, the resources, $Resource_{req}$, of an organization, the authorization module will determine the policies that are applicable, *AP*, for the request, i.e. $AP \subseteq RP_i \subseteq PS_i \subseteq PS$. Hence, any elements of PS, PS_i, and RP_i might be applicable to a given request; with each having conflicting authorization decision.

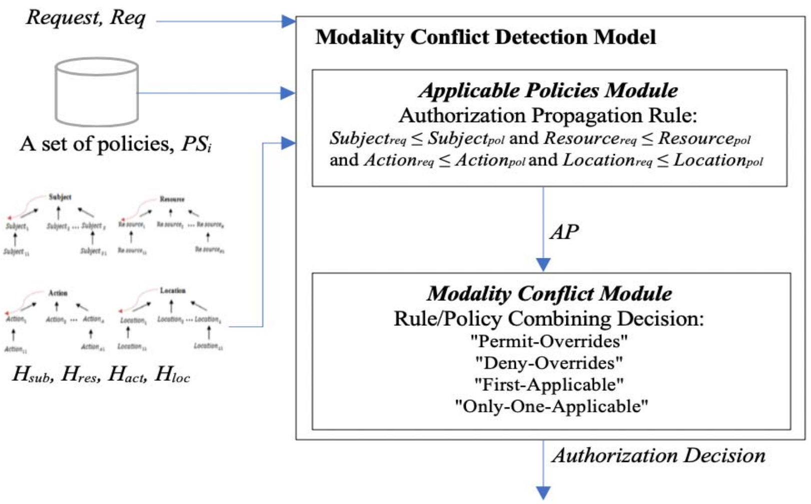

Request, Req
A set of policies, PS_i
Subject
Resource
Action
Location
H_{sub}, H_{res}, H_{act}, H_{loc}
Modality Conflict Detection Model
Applicable Policies Module
Authorization Propagation Rule:
$Subject_{req} \le Subject_{pol}$ and $Resource_{req} \le Resource_{pol}$ and $Action_{req} \le Action_{pol}$ and $Location_{req} \le Location_{pol}$
AP
Modality Conflict Module
Rule/Policy Combining Decision:
"Permit-Overrides"
"Deny-Overrides"
"First-Applicable"
"Only-One-Applicable"
Authorization Decision

Figure 1: The Modality Conflict Detection Model

Table 1: Examples of the University Policies

Policy Combining Algorithm	Rule Combining Algorithm	Policy No.	Effect	Subject	Resource	Action	Condition
Permit-Overrides	Permit-Overrides	*Pol*1	Permit	RA	ExternalGrades	Assign ∨ View	(Location = Association) ∧ (Time ≥ 12P.M. ∧ Time ≤ 2P.M.) ∧ (Email = upm.edu.my)
Permit-Overrides	Permit-Overrides	*Pol*2	Deny	Student	Course	Assign ∨ View	(Location = Department) ∧ (Time ≥ 12P.M. ∧ Time ≤ 1P.M.) ∧ (Email = upm.edu.my)
		*Pol*3	Permit	Undergrad	Course	View	(Location = Department) ∧ (Time ≥ 12P.M. ∧ Time ≤ 1P.M.) ∧ (Email = upm.edu.my)
Deny-Overrides	Permit-Overrides	*Pol*4	Permit	AssociateProfessor	Grades	Assign ∨ View ∨ SubmitGrade ∨ SubmitGradeChange	(Location = GraduateSchool) ∧ (Time ≥ 12P.M. ∧ Time ≤ 1P.M.)
	Permit-Overrides	*Pol*5	Deny	Faculty_Member	Grades	Assign ∨ View	(Location = School) ∧ (Time ≥ 12P.M. ∧ Time ≤ 1P.M.)

Table 2: Examples of Requests

Request No.	Subject	Resource	Action	Condition
*Req*1	Undergraduate Student	Teaching Course	View	(Location = University Department) ∧ (Time = 12.30P.M.) ∧ (Email = gs23442@upm.edu.my)
*Req*2	ResearchAssistant	ExternalGrades	Assign	(Location = Institute) ∧ (Time = 1.30P.M.) ∧ (Email = gs23442@upm.edu.my)
*Req*3	AssociateProfessor	InternalGrades	Assign	(Location = GraduateSchool) ∧ (Time = 12.30P.M.) ∧ (Email = gs23442@upm.edu.my)
*Req*4	Faculty_Member	Grades	View	(Location = School) ∧ (Time = 12.30P.M.)
*Req*5	AssociateProf	Grades	Assign	(Location = GraduateSchool) ∧ (Time = 12.30P.M.)
*Req*6	Faculty_Member	Grades	AssignGrade	(Location = School) ∧ (Time = 12.30P.M.)

XACML policy language supports policy or rule combination algorithms, which evaluate the applicable policy based on the logic of the algorithm. However, current authorization modules do not support propagation rule by which an authorization can be granted on the basis of the presence or absence of other authorizations. Our *applicable policies module* employs the subject hierarchy (H_{sub}), resource hierarchy (H_{res}), action hierarchy (H_{act}), and location hierarchy (H_{loc}) to support a more adequate representation of their semantics. Figure 2 depicts an example of H_{sub}, H_{res}, H_{act}, and H_{loc} for the university policies that are formed based on the results collected from human experts.

From the authorizations that are explicitly specified, the module aims to automatically derive all possible implicit authorizations for a given request, i.e. $AP = AP_{explicit} \cup AP_{implicit}$ where $AP_{explicit}$ is the set of applicable policies based on the explicit policies, i.e. $AP_{explicit} \subseteq RP_i \subseteq PS_i \subseteq PS$, while $AP_{implicit}$ is the set of applicable policies derived based on the explicit policies, $AP_{explicit}$. The following gives the definitions of explicit and implicit authorizations as used in this paper:

Definition 1 When an access right for a subject on a resource is explicitly specified in the access control policy, this is referred to as *explicit authorization* [6].

Definition 2 When an access right for a subject on a resource can be implicitly derived from other explicit authorizations, this is referred to as *implicit authorization* [6].

With authorization propagation, a policy defined on a parent node, $P_{i\text{-}parent}$, is automatically propagated along the subject, resource, action, and location hierarchies to all its child nodes. Each child node either has its own policy, $P_{j\text{-}child}$, that represents an explicit policy or inherits the policy from its parent, $P_{i\text{-}parent}$, that now represents an implicit policy to the child node. A conflict authorization decision occurs when both policies $P_{j\text{-}child}$ and $P_{i\text{-}parent}$ have conflicting decision, i.e. permit and deny decisions. Thus, a user may be permitted to access a resource if one access hierarchy path is selected, and deny through another path [26].

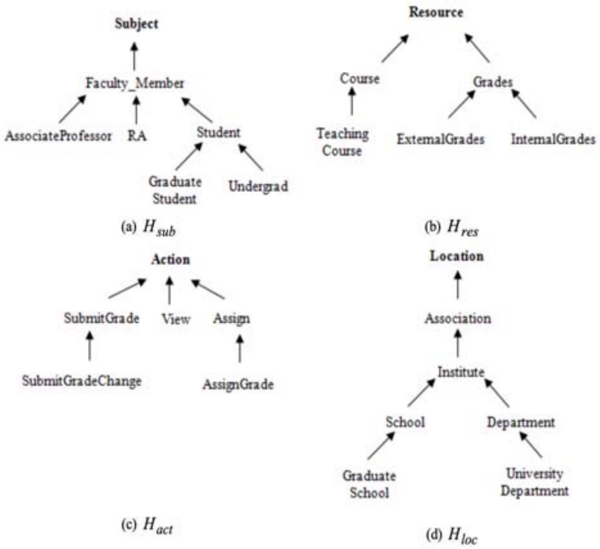


Figure 2: (a) Subject Hierarchy, H_{sub}, (b) Resource Hierarchy, H_{res}, (c) Action Hierarchy, H_{act}, and (d) Location Hierarchy, H_{loc}, for the University Example

In identifying an applicable policy, the semantic relationships (class-subclass) among the subject, resource, action and location of a request and a policy are analyzed. We provide an authorization propagation rule which is defined as follows:

$Subject_{req} \leq Subject_{pol}$ and $Resource_{req} \leq Resource_{pol}$ and $Action_{req} \leq Action_{pol}$ and $Location_{req} \leq Location_{pol}$ (Rule 1)

where $Subject_{req}$ ($Subject_{pol}$) is the subject of the request (policy, respectively), $Resource_{req}$ ($Resource_{pol}$) is the resource of the request (policy, respectively), $Action_{req}$ ($Action_{pol}$) is the action of the request (policy, respectively), $Location_{req}$ ($Location_{pol}$) is the location of the request (policy, respectively).

The concept is classified as a partial ordered structure where an attribute value of a request, av_{req} is a specialization of an attribute value of a policy, av_{pol} if and only if $av_{req} \leq av_{pol}$, where $\leq$ represents the subsumption operator. This structure can ground the permission inheritance of the authorization propagation, i.e. rights assigned to concepts can be inherited by subsumed concepts [9]. The underlying idea is that the parent-child relationship implies that one rule could be a restriction of the other and this would be more helpful than the sibling relationship [21]. Figure 3 presents the *applicable policies algorithm*.

```
Input:  Req($Subject_{req}$, $Resource_{req}$, $Action_{req}$,
Location_req) – A request from a user in $Org_A$; $PS_i$ – A set of policies, $\{P_1, P_2, \ldots, P_m\}$ in $Org_B$ with $P_i(Subject_{pol}, Resource_{pol}, Action_{pol}, Location_{pol})$; Hierarchy – Subject hierarchy, $H_{sub}$, resource hierarchy, $H_{res}$, action hierarchy, $H_{act}$, location hierarchy, $H_{loc}$
Output: A set of applicable policies, AP
1. Begin
2.   For each $P_i \in PS_i$ do
3.   Begin
4.     Apply N-gram similarity measure and WordNet
       between: $(Subject_{req}, Subject_{pol})$, $(Resource_{req}, Resource_{pol})$, $(Action_{req}, Action_{pol})$, and $(Location_{req}, Location_{pol})$
5.     If $(Subject_{req} \leq Subject_{pol})$ && $(Resource_{req} \leq Resource_{pol})$ && $(Action_{req} \leq Action_{pol})$ &&
          $(Location_{req} \leq Location_{pol})$
       Then $AP = AP \cup P_i$
6.   End
7.   Return AP
8. End
```

Figure 3: The Applicable Policies Algorithm

The attribute value of a request, *Req*, is compared to the attribute value of each policy, $P_i \in PS_i$, to identify the semantic relationships between them (line 2). In matching the attribute values of a request and a policy, *N-gram* and WordNet as an external thesaurus are employed (line 4). *N*-gram is utilized to resolve the syntactic variations while WordNet is utilized to resolve the terminological variations. Based on the proposed authorization propagation rule, the explicit and implicit policies will be retrieved if the above propagation rule conditions are meet (line 5).

Table 3 presents the explicit and implicit applicable policies retrieved for each of the request given in Table 2. The proposed solution is able to retrieve the same explicit and implicit applicable policies as retrieved by the human experts for all the requests. For example, referring to the *Req*1, the implicit policies, $Pol2_{implicit}$, is derived from *Pol*2. *Undergraduate Student* in *Req*1 is matched to *Undergrad* based on the *N-gram* similarity measure and WordNet and *Undergrad* is a child node of *Student* in the subject hierarchy as shown in Figure 2(a). Hence, *Undergraduate Student* in *Req*1 is a subclass of *Student* in *Pol*2. *Teaching Course* in *Req*1 is a subclass of *Course* in *Pol*2 based on Figure 2(b). *View* in *Req*1 is an exact matched of *View* in *Pol*2. *University Department* in *Req*1

Table 3: Applicable Policy Retrieved and Modality Conflict Detection Results of the Sun's XACML Implementation, Proposed Solution, and Human Experts

Engine	Request	Explicit Policy	Implicit Policy	Modality Conflict		Combining Algorithm	Authorization Decision
				Policy Level	Policy Set Level		
Sun's XACML Implementation	*Req*1	-	-	No	No	-	N/A
	*Req*2	-	-	No	No	-	N/A
	*Req*3	-	-	No	No	-	N/A
	*Req*4	*Pol*5	-	No	No	-	Deny
	*Req*5	-	-	No	No	-	N/A
	*Req*6	-	-	No	No	-	N/A
Proposed Solution	*Req*1	*Pol*2 *Pol*3	$Pol2_{implicit}$, $Pol3_{implicit}$	Yes	No	Permit-Overrides	Permit
	*Req*2	*Pol*1	$Pol1_{implicit}$	No	No	-	Permit
	*Req*3	*Pol*4 *Pol*5	$Pol4_{implicit}$, $Pol5_{implicit}$	No	Yes	Deny-Overrides	Deny
	*Req*4	*Pol*4 *Pol*5	-	No	No	-	Deny
	*Req*5	*Pol*4 *Pol*5	$Pol4_{implicit}$, $Pol5_{implicit}$	No	Yes	Deny-Overrides	Deny
	*Req*6	*Pol*4 *Pol*5	$Pol5_{implicit}$	No	No	-	Deny
Human Expert	*Req*1	*Pol*2 *Pol*3	$Pol2_{implicit}$, $Pol3_{implicit}$	Yes	No	Permit-Overrides	Permit
	*Req*2	*Pol*1	$Pol1_{implicit}$	No	No	-	Permit
	*Req*3	*Pol*4 *Pol*5	$Pol4_{implicit}$, $Pol5_{implicit}$	No	Yes	Deny-Overrides	Deny
	*Req*4	*Pol*4 *Pol*5	-	No	No	-	Deny
	*Req*5	*Pol*4 *Pol*5	$Pol4_{implicit}$, $Pol5_{implicit}$	No	Yes	Deny-Overrides	Deny
	*Req*6	*Pol*4 *Pol*5	$Pol5_{implicit}$	No	No	-	Deny

is a subclass of *Department* in *Pol*2 since *University Department* is a child node of *Department* based on Figure 2(d). Each implicit policy shown in Table 3 is derived using similar process as explained above.

3.2. Detecting Modality Conflict

The *applicable policies module* may derive more than one applicable policy with modalities of opposite effects. The modality conflict could arise from semantic relationships between concepts that cannot be detected simply by looking at the terminology structure of the terms. Definition 3 defined the modality conflict as used in this paper.

Definition 3 Modality conflict is inconsistency in the restriction policy specification, which arises when two policies *Pol*1 and *Pol*2 with modalities of opposite effects for the same subject, resource, action, and condition, i.e.:

*Pol*1 ≡ (Permit, $Subject_{pol}$, $Resource_{pol}$, $Action_{pol}$, $Condition_{pol}$)
*Pol*2 ≡ (Deny, $Subject_{pol}$, $Resource_{pol}$, $Action_{pol}$, $Condition_{pol}$)

The *modality conflict algorithm* as shown in Figure 4 checks whether modality conflict exists among the applicable policies and if it occurs then conflict resolution is needed to resolve the conflict before an authorization decision is returned (line 4). XACML defines four types of predefined combining algorithm to automatically resolve modality conflict, namely: Permit-Overrides, Deny-Overrides, First-Applicable, and Only-One-Applicable. Table 3 presents the modality conflict detected at different policy level.

```
Input:  A set of applicable policies, $AP = \{AP_1, AP_2, …, AP_l\}$ with $AP_i ≡ (Effect_{Pi}, Subject_{Pi}, Resource_{Pi}, Action_{Pi}, Condition_{Pi})$
Output:  Conflict/No Conflict
1.  Begin
2.    For each $AP_i \in AP$
3.      For each $AP_j \in AP$
4.        If $Effect_{Pi} \neq Effect_{Pj}$ at policy set level
            Return Conflict
            Resolve modality conflict based on the policy combining algorithm as stated in the policy
          ElseIf $Effect_{Pi} \neq Effect_{Pj}$ at policy level
            Return Conflict
            Resolve modality conflict based on the rule combining algorithm as stated in the rule
          Else Return No Conflict
5.  End
```

Figure 4: The Modality Conflict Algorithm

*Req*1, *Req*3, and *Req*5 are the requests in which modality conflict is detected by the human experts as well as our proposed solution, as explained below:

i. For *Req*1, the proposed solution applied the rule combining algorithm, "Permit-Overrides" to resolve the modality conflict, in which "Permit" is returned as the authorization decision. For *Req*3 and *Req*5, the proposed solution chooses the policy combining algorithm, "Deny-Overrides" to resolve the modality conflict at the policy set level, in which "Deny" is returned as the authorization decision. While Sun's XACML implementation did not detect any modality conflict for *Req*1, *Req*3, and *Req*5, thus, "N/A" is returned as the authorization decision.
ii. For *Req*2 and *Req*6, there is no modality conflict detected by the proposed solution. Thus, the effect of $Pol1_{implicit}$, "Permit" and the effect of $Pol5_{implicit}$, "Deny" are returned as the authorization decision for *Req*2 and *Req*6, respectively. "N/A" is returned as the authorization decision by Sun's XACML implementation since there is no applicable policy retrieved for *Req*2 and *Req*6.
iii. For *Req*4, both the Sun's XACML implementation and the proposed solution retrieved the explicit policy, *Pol*5 and the effect of *Pol*5, "Deny" is returned as the authorization decision.

4. RESULTS AND DISCUSSION

We have designed and performed four analyses with the following aims: (i) to measure the accuracy of the proposed modality conflict detection model in identifying the applicable policies as opposed to the Sun's XACML implementation and the human expert results, (ii) to show that the attributes considered in this study can affect the decision as to whether the applicable policies should be retrieved or not, (iii) to measure the accuracy of the proposed modality conflict detection model with respect to the modality conflict detected as opposed to the previous method and the human expert results, and (iv) to show that the attributes considered in this study can affect the accuracy of the modality conflict detection. These analyses are performed on a laptop running Window 7 Home Premium Service Pack 1 operating system with 6 GB of physical memory and 2.50GHz Intel(R) Core (TM) i5-3210M machine. The policies and requests are presented based on the syntax and structure of XACML.

This work used six sets of XACML policies[2] that have been designed for a university and a conference management domain, namely: *CodeA, CodeB, CodeC, CodeD, Continue-a*, and *Continue-b*. These XACML policies are taken from [22]. *Continue-a* and *Continue-b* are designed for a conference management while *CodeA*, *CodeB*, *CodeC*, and *CodeD* are designed for a real-world web application supporting the university domain. Another two sets of policies that have been analyzed are taken from [40]. Those policies are based on the RBAC model and are designed for a university[3] and a health care institution[4]. The RBAC policies are presented in the syntax and structure of XACML with positive effects since negative authorizations are not supported in the RBAC model [37]. The modality conflict only occurs between the positive and negative authorization policies [37]. Only two of the policies mentioned above can be further used for modality conflict detection, which are *Continue-a* and *Continue-b*. The *CodeA*, *CodeB*, *CodeC*, and *CodeD* and the RBAC policies are modified by adding additional deny rule for each policy since these policies do not contain negative authorization policy. Besides that, these sets of policies are modified by adding additional condition context since initially these policies do not contain condition context.

We used the request-generation technique [24] to generate a random of 10,000 requests since most of the real-world systems use much less than 10,000 policies [26]. Eight sets of the modified XACML policy datasets are used as the source to generate the random requests. Owing to the fact that there is a distinct lack of real request datasets in distributed environment, therefore the concept nodes from the Univ-Bench[5] ontology are selected. This university ontology has been used as benchmark tests in the Semantic Web and Agent Technologies (SWAT)[6] projects [15]. Besides that, the Semantic Web Conference (SWC)[7] ontology is also selected. SWC ontology is developed to support the European Semantic Web Conference; while Logical Observation Identifiers Names and Codes (LOINC) clinical document ontology [8], a database and universal standard for identifying medical laboratory observations in health-care institution is also selected. Since the real policy datasets used in our work are based on the university, conference management, and health-care institution domain, the domain ontologies which are related to these three domains are selected as the source to generate the random requests. These domain ontologies are obtained from SWOOGLE ontology search engine[9].

In order to measure the accuracy of the proposed solution, Precision (*P*), Recall (*R*), and F-Measure (*F*) originated from the information retrieval field are used [12]; as presented below:

$$\text{Precision } (P) = \frac{|TP|}{|FP| + |TP|} \quad (1)$$

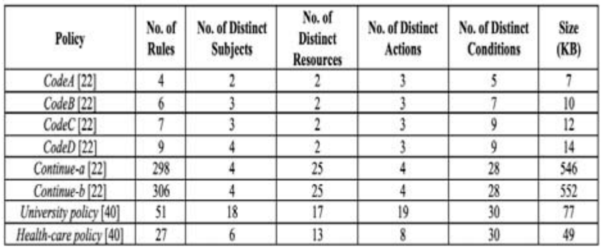
$$\text{Recall } (R) = \frac{|TP|}{|FN| + |TP|} \quad (2)$$

$$\text{F-Measure } (F) = \frac{2}{\frac{1}{R} + \frac{1}{P}} \quad (3)$$

where *TP* is a true positive when a case is positive and predicted positive, *FN* is a false negative when a case is positive but predicted negative, while *FP* is a false positive when a case is negative but predicted positive. *P* varies in the [0, 1] range; the higher the value, the smaller is the set of wrong mappings (false positives) that has been computed. It is a correctness measure. *R* varies in the [0,1] range; the higher the value, the smaller is the set of correct mappings (true positives) that is not found. It is a completeness measure. *F* varies in the [0,1] range. The version computed here is the harmonic mean of precision and recall. Table 4 shows the characteristics of the modified XACML policy datasets, while Table 5 shows the characteristics of the XACML request datasets.

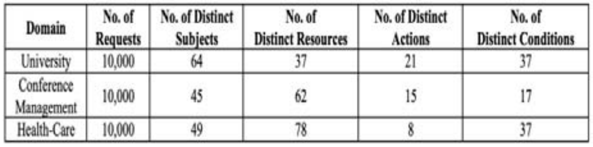
[2] http://sourceforge.net/projects/xacmlpdp/
[3] http://www3.cs.stonybrook.edu/~stoller/ccs2007/university-policy.txt
[4] http://www3.cs.stonybrook.edu/~stoller/ccs2007/healthcare.txt
[5] http://swat.cse.lehigh.edu/onto/univ-bench.owl
[6] http://swat.cse.lehigh.edu/projects/lubm/
[7] http://data.semanticweb.org/ns/swc/swc_2009-05-09.html
[8] https://loinc.org/discussion-documents/document-ontology/loinc-document-ontology-axis values?force_toc:int=1
[9] http://swoogle.umbc.edu/

Table 4: The Main Characteristics of the Modified XACML Policy Datasets



Policy	No. of Rules	No. of Distinct Subjects	No. of Distinct Resources	No. of Distinct Actions	No. of Distinct Conditions	Size (KB)
CodeA [22]	4	2	2	3	5	7
CodeB [22]	6	3	2	3	7	10
CodeC [22]	7	3	2	3	9	12
CodeD [22]	9	4	2	3	9	14
Continue-a [22]	298	4	25	4	28	546
Continue-b [22]	306	4	25	4	28	552
University policy [40]	51	18	17	19	30	77
Health-care policy [40]	27	6	13	8	30	49

Table 5: The Main Characteristics of the XACML Request Datasets



Domain	No. of Requests	No. of Distinct Subjects	No. of Distinct Resources	No. of Distinct Actions	No. of Distinct Conditions
University	10,000	64	37	21	37
Conference Management	10,000	45	62	15	17
Health-Care	10,000	49	78	8	37

In terms of accuracy, we compared the applicable policies identified and the modality conflict detected among the applicable policies by our proposed solution to those obtained by Sun's XACML implementation [31] and the human experts for each request. We choose Sun's XACML implementation in our comparison for two reasons. First, it is the first and the most widely deployed XACML evaluation engine and has become the industrial practice [22]. Second, the previous works [3, 22, 29] selected Sun's XACML implementation for their results comparison since Sun's XACML implementation is an open source. These works focused on the efficiency of their engine by reducing the processing time while the results obtained are the same as compared to Proctor [31]. While our work focuses on the accuracy of identifying the applicable policies and detecting modality conflict among the applicable policies.

To provide a ground for evaluating the quality of the matching results, the results produced by our proposed solutions are compared to the human experts' results. The task is first conducted manually by three professional human experts who are either familiar with database management or English linguistics. Each experiment is repeated for 5 times. The final results are the average of these 5 running experiments. The results of *P*, *R*, and *F* in retrieving the applicable policies and in detecting modality conflict by the proposed solution and the Sun's XACML implementation are compared to the real match results obtained by the human experts. The results are analyzed at various values of similarity thresholds. A higher value of similarity threshold implies stricter matching requirements between the string elements of a request and a policy. A higher value of *P*, *R*, and *F* implies a higher value of accuracy achieved by the solution.

4.1 Experiment on Applicable Policies Identification

This experiment aims to measure the accuracy of the proposed modality conflict detection model in identifying the applicable policies as opposed to the previous method with the real results by the human experts. The results of *P*, *R*, and *F* of this experiment are presented in figures 5, 6, and 7, respectively.

Based on Figure 5, we observed that the *P* obtained by the proposed solution is higher when the similarity threshold is set to a higher value. While the Sun's XACML implementation also obtained perfect *P* in retrieving the applicable policies because all the results returned are true positives which are the same results as those produced by the human experts.

Based on Figure 6 and Figure 7, we observed that the *R* and *F* achieved by the proposed solution are higher than the *R* and *F* achieved by the Sun's XACML implementation. The proposed solution is able to identify the implicit applicable policies based on the proposed authorization propagation, hence outperforms the Sun's XACML implementation in terms of *R* and *F* for all sets of policies.

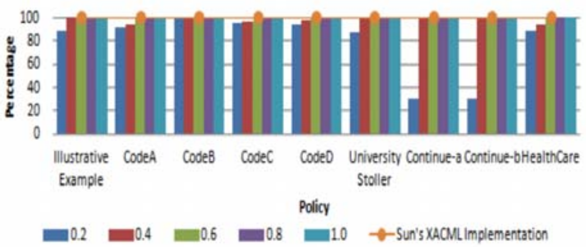


Figure 5: Precision (P) of the Proposed Solution with Different Similarity Thresholds and Sun's XACML Implementation in Retrieving the Applicable Policies

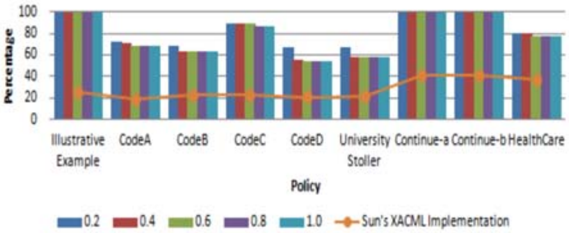


Figure 6: Recall (R) of the Proposed Solution with Different Similarity Thresholds and Sun's XACML Implementation in Retrieving the Applicable Policies

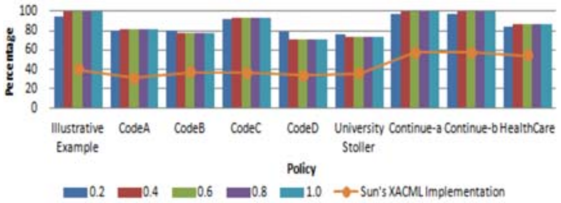


Figure 7: F-measure (F) of the Proposed Solution with Different Similarity Thresholds and Sun's XACML Implementation in Retrieving the Applicable Policies

4.2 Experiment on Applicable Policies Identification with and without Condition Attribute

This experiment aims to show that the attributes considered in this study can affect the decision as to whether the applicable policies will be retrieved or not which further affect the accuracy of the modality conflict detection. To perform this, we compare the solution which only considered three attributes (subject, resource, and action) to the solution which considered four attributes (subject, resource, action, and condition). Table 6 presents the results of *P*, *R*, and *F* of this experiment.

Based on the Table 6, we observed that by considering the spatial and temporal constraints, the solution results in higher percentage of *P* and *F* for all sets of policies but lower percentage of *R* compared to when these constraints are not considered. Without considering the condition attribute, the number of false positives produced and the possibility to retrieve irrelevant policies is higher than the solution with the condition attribute.

However, the solution without condition attribute achieved higher percentage of *R* compared to when these constraints are considered. The solution with condition attribute produced higher number of false negatives in retrieving the applicable policies than the solution without condition attribute as it has to perform similar matching process in which the condition is being considered. Hence, the applicable policy will not be retrieved if one of the subjects, resources, actions, or conditions of a request and a policy is mismatch.

Table 6: Precision, (P), Recall, (R), and F-measure, (F) with and without the Condition Attribute based on Different Similarity Thresholds in Retrieving the Applicable Policies

Policy	Evaluation Metric	Without Condition Attribute					With Condition Attribute				
		0.2	**0.4**	**0.6**	**0.8**	**1.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
CodeA	**Precision (*P*)**	65.68	77.21	82.83	82.83	82.83	91.30	94.44	100	100	100
	Recall (*R*)	84.52	81.62	75.53	75.53	75.53	72.41	71.20	68.63	68.63	68.63
	F-Measure (*F*)	73.92	78.87	79.01	79.01	79.01	80.77	81.19	81.40	81.40	81.40
CodeB	**Precision (*P*)**	86.90	74.43	74.43	74.43	74.43	98.95	100	100	100	100
	Recall (*R*)	77.16	74.48	74.48	74.48	74.48	67.74	63.50	63.50	63.50	63.50
	F-Measure (*F*)	81.74	74.45	74.45	74.45	74.45	80.77	77.68	77.68	77.68	77.68
CodeC	**Precision (*P*)**	75.73	74.26	74.26	74.26	74.26	95.03	96.75	96.75	100	100
	Recall (*R*)	91.14	92.14	92.14	92.14	92.14	89.47	89.10	89.10	87.05	87.05
	F-Measure (*F*)	82.72	82.23	91.14	91.14	91.14	92.17	92.77	92.77	93.08	93.08
CodeD	**Precision (*P*)**	76.51	70.31	70.31	70.31	70.31	94.63	97.94	100	100	100
	Recall (*R*)	71.86	65.24	65.24	65.24	65.24	67.46	55.29	54.29	54.29	54.29
	F-Measure (*F*)	74.11	67.68	67.68	67.68	67.68	78.77	70.68	70.37	70.37	70.37
UniversityStoller	**Precision (*P*)**	64.17	89.98	89.98	89.98	89.98	87.94	100	100	100	100
	Recall (*R*)	69.16	59.73	58.73	58.73	58.73	67.05	57.99	57.99	57.99	57.99
	F-Measure (*F*)	66.57	71.80	71.07	71.07	71.07	76.09	73.41	73.41	73.41	73.41
Continue-a	**Precision (*P*)**	20.79	79.18	98.12	98.12	98.12	30.21	100	100	100	100
	Recall (*R*)	99.94	99.72	99.65	99.65	99.65	99.41	99.35	99.35	99.35	99.35
	F-Measure (*F*)	34.41	88.27	98.88	98.88	98.88	96.91	99.67	99.67	99.67	99.67
Continue-b	**Precision (*P*)**	20.79	79.18	98.12	98.12	98.12	30.21	100	100	100	100
	Recall (*R*)	99.79	99.72	99.65	99.65	99.65	99.35	99.32	99.32	99.32	99.32
	F-Measure (*F*)	34.41	88.27	98.88	98.88	98.88	96.87	99.65	99.65	99.65	99.65
HealthCare	**Precision (*P*)**	79.18	97.90	97.90	97.90	97.90	89.34	94.78	100	100	100
	Recall (*R*)	84.72	80.74	80.74	80.74	80.74	79.56	79.56	77.10	77.10	77.10
	F-Measure (*F*)	81.86	88.50	88.50	88.50	88.50	84.17	86.51	87.07	87.07	87.07

4.3 Experiment on Modality Conflict Detection

This experiment aims to measure the accuracy of the proposed modality conflict detection model with respect to the modality conflict detected as opposed to the previous method and the human expert results. The applicable policies which are retrieved for a request can affect the decision whether modality conflict occurs among the applicable policies and the modality conflict is resolved based on the combining algorithm before the final decision is returned. The results of *P*, *R*, and *F* of this experiment are presented in figures 8, 9, and 10, respectively.

Sun's XACML implementation achieved 0% *P*, 0% *R*, and 0% *F* in detecting the modality conflict for *CodeA*, *CodeB*, *CodeC*, *CodeD*, *Continue-a*, *Continue-b*, and *HealthCare*. Even for *UniversityStoller* policy, the Sun's XACML implementation achieved lower percentage of *P*, *R*, and *F* in detecting the modality conflict compared to the proposed solution. This is due to the Sun's XACML implementation produced the maximum value of *P* with no false positive in retrieving the applicable policies for all sets of policies. However, the Sun's XACML implementation produced poor *R* and *F* in retrieving the applicable policies which caused the Sun's XACML implementation achieved poor *R* and *F* in detecting the modality conflict for all sets of policies.

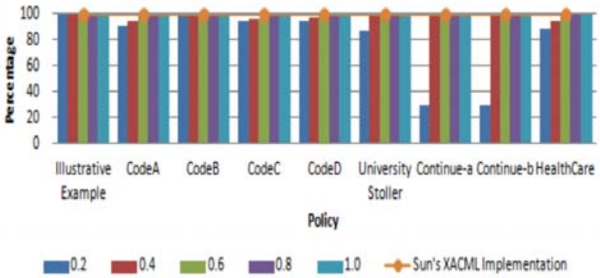


Figure 8: Precision (P) for Modality Conflict Detection by the Proposed Solution with Different Similarity Thresholds and Sun's XACML Implementation

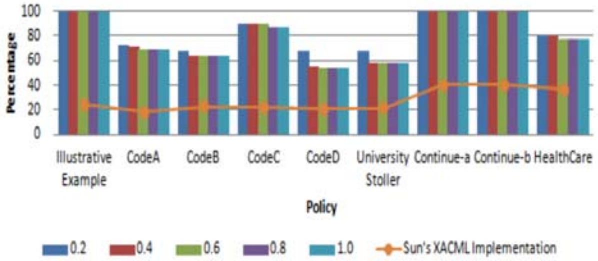


Figure 9: Recall (R) for Modality Conflict Detection by the Proposed Solution with Different Similarity Thresholds and Sun's XACML Implementation

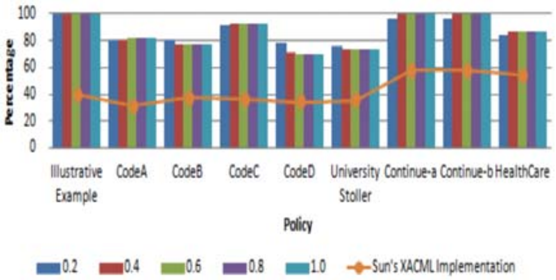


Figure 10: F-measure (F) for Modality Conflict Detection by the Proposed Solution with Different Similarity Thresholds and Sun's XACML Implementation

We observed that the proposed solution achieved the lowest percentage of *P* in detecting the modality conflict since the proposed solution achieved the lowest percentage of *P* in retrieving the applicable policies for *continue-a* and *continue-b* policies (Figure 5). This is because both policies contained the largest size of policies among the datasets, and hence, the number of false positive in retrieving the applicable policies increased. For example, for *Continue-a*, the proposed solution achieved only 30.21% *P* in retrieving the applicable policies and 20.66% *P* in detecting the modality conflict when the similarity threshold is set to 0.2. This finding corroborates the concept that if one of the subjects, resources, actions, or conditions of a request and a policy is mismatch, the applicable policy will not be retrieved and further effects the modality conflict detection.

Since our proposed solution achieved higher value of *R* and *F* in retrieving the applicable policies when the similarity threshold is set to a higher value, it also achieved higher value of *R* and *F* in detecting the modality conflict when the similarity threshold is set to a higher value compared to the Sun's XACML implementation for all sets of policies.

4.4 Experiment on Modality Conflict Detection with and without Condition Attribute

This experiment is designed and performed to show the impact of considering the condition attributes in detecting the modality conflict. To perform this, we compare the solution which only considered three attributes (subject, resource, and action) to the solution which considered four attributes (subject, resource, action, and condition). The results of *P*, *R*, and *F* of this experiment are presented in Table 7.

Table 7: Precision (P), Recall (R), and F-measure (F) with and without the Condition Attribute based on Different Similarity Thresholds in Detecting the Modality Conflict

Policy	Evaluation Metric	Without Condition Attribute					With Condition Attribute				
		0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0
CodeA	**Precision (*P*)**	48.00	66.37	70.00	70.00	70.00	60.00	75.00	100	100	100
	Recall (*R*)	30.44	29.65	29.65	29.65	29.65	27.27	27.27	27.27	27.27	27.27
	F-Measure (*F*)	37.25	40.99	41.66	41.66	41.66	37.50	40.00	42.86	42.86	42.86
CodeB	**Precision (*P*)**	78.33	78.33	78.33	78.33	78.33	100	100	100	100	100
	Recall (*R*)	33.00	33.00	33.00	33.00	33.00	32.43	32.43	32.43	32.43	32.43
	F-Measure (*F*)	46.44	46.44	46.44	46.44	46.44	48.98	48.98	48.98	48.98	48.98
CodeC	**Precision (*P*)**	77.99	77.99	77.99	85.00	85.00	90.48	90.48	96.48	100	100
	Recall (*R*)	81.14	82.56	82.56	82.56	82.56	80.85	80.85	80.85	80.85	80.85
	F-Measure (*F*)	79.53	80.21	80.21	80.21	80.21	85.39	85.39	85.39	89.41	89.41
CodeD	**Precision (*P*)**	83.51	79.39	79.39	79.39	79.39	100	100	100	100	100
	Recall (*R*)	36.01	18.22	19.22	19.22	19.22	35.48	17.24	17.24	17.24	17.24
	F-Measure (*F*)	50.32	29.63	29.63	29.63	29.63	52.38	29.41	29.41	29.41	29.41
UniversityStoller	**Precision (*P*)**	43.33	65.78	65.78	65.78	65.78	72.73	100	100	100	100
	Recall (*R*)	69.05	69.05	69.05	69.05	69.05	67.05	57.99	57.99	57.99	57.99
	F-Measure (*F*)	53.25	67.38	67.38	67.38	67.38	76.09	73.41	73.41	73.41	73.41
Continue-a	**Precision (*P*)**	11.46	69.78	88.39	88.39	88.39	20.66	100	100	100	100
	Recall (*R*)	84.44	75.72	75.72	75.72	75.72	78.13	75.00	75.00	75.00	75.00
	F-Measure (*F*)	20.18	72.63	72.63	72.63	72.63	32.68	85.71	85.71	85.71	85.71
Continue-b	**Precision (*P*)**	11.46	69.78	88.39	88.39	88.39	20.66	100	100	100	100
	Recall (*R*)	80.21	75.72	75.72	75.72	75.72	75.00	75.00	75.00	75.00	75.00
	F-Measure (*F*)	20.05	72.63	72.63	72.63	72.63	31.58	85.71	85.71	85.71	85.71
HealthCare	**Precision (*P*)**	63.27	90.00	90.00	90.00	90.00	76.92	100	100	100	100
	Recall (*R*)	37.77	32.23	32.23	32.23	32.23	34.48	32.14	32.14	32.14	32.14
	F-Measure (*F*)	47.30	47.46	47.46	47.46	47.46	47.62	48.65	48.65	48.65	48.65

We observed that the solution without the condition attribute achieved lower percentage of *P* and *F* in detecting the modality conflict for all sets of policies but higher percentage of *R* compared to the solution with the condition attribute. Hence, the solution that does not consider the temporal and spatial attributes produced lower number of false negatives but higher number of false positives than the solution that considered these constraints, since most of the applicable policies are not returned by the solution. This indicates that the solution with condition attribute is better compared to the solution without the condition attribute.

5. CONCLUSION

Policy evaluation has received considerable attention to accommodate the security requirements covering large, open, distributed, and heterogeneous computing environments. This research addresses the significant need in identifying the applicable policies and detecting modality conflict for XACML policy evaluation. Our modality conflict model supports the authorization propagation rule which explores inheritance relationships of a subject, resource, action, and condition which enables the applicable policies to be retrieved for a given request. We present the algorithms for identifying applicable policies and detecting modality conflict based on the proposed authorization propagation rule.

The experimental results show that our proposed solution achieved lower percentage of *R* but higher percentage of *P* and *F* for all sets of policies when more attributes are considered in retrieving the applicable policies and in detecting the modality conflict compared when these constraints are not considered. Besides that, our proposed solution achieved higher percentage of *P*, *R* and *F* in retrieving the applicable policies and in detecting modality conflict as compared to the previous work. The accuracy of the proposed solution indicates that our proposed solution is better than the Sun's XACML implementation in policy evaluation.

The proposed solution can be further enhanced by considering other factors which could affect the authorization decisions such as obligations for which some actions should be launched once certain conditions are satisfied. Further enhancement to the proposed solution in this area can be done by investigating the spatial context of a request and a policy which is organized based on the logical data model for used by the geographic information system (GIS).

REFERENCES:

[1] Adi, K., Bouzida, Y., Hattak, I., Logrippo, L., & Mankovskii, S., "Typing for Conflict Detection in Access Control Policies", *Proceedings of the 4th International Conference on E-Technologies* (*MCETECH*), 2009, pp. 212-226.

[2] Almutairi, A., Sarfraz, M., Basalamah, S., Aref, W., & Ghafoor, A., "A Distributed Access Control Architecture for Cloud Computing", *IEEE Software*, Vol. 29, No. 2, 2012, pp. 36-44.

[3] Ammar, N., Malik, Z., Bertino, E., & Rezgui, A., "XACML Policy Evaluation with Dynamic Context Handling", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 9, 2015, pp. 2575-2588.

[4] Bertino, E., Brodie, C., Calo, S. B., Cranor, L. F., Karat, C., Karat, J., Li, N., Lin, D., Lobo, J., Ni, Q., & Rao, P. R., "Analysis of Privacy and Security Policies", *IBM Journal of Research and Development*, Vol. 5, No. 2, 2009, pp. 3:1-3:18.

[5] Bertino, E., Buccafurri, F., Ferrari, E., & Rullo, P., "An Authorization Model and its Formal Semantics", *Proceedings of the 5th European Symposium on Research in Computer Security* (*ESORICS*), 1998, pp. 127-142.

[6] Bertino, E., Ghinita, G., & Kamra, A., "Access Control for Databases: Concepts and Systems", *Foundations and Trends in Databases*, Vol. 3, No. 1-2, 2011, pp. 1-148.

[7] Boutaba, R. & Aib, I., "Policy Based Management: A Historical Perspective", *Journal of Network and Systems Management*, Vol. 15, No. 4, 2007, pp. 447-480.

[8] Brodecki, B., Szychowiak, M., & Sasak, P., "Security Policy Conflicts in Service Oriented Systems", *New Generation Computing*, Vol. 30, No. 2-3, 2012, pp. 215-240.

[9] Damiani, E., di Vimercati, S. D. C., Fugazza, C., & Samarati, P., "Modality Conflicts in Semantics Aware Access Control", *Proceedings of the 6th International Conference on Web Engineering* (*ICWE*), 2006, pp. 249-256.

[10] Damianou, N., Bandara, A., Sloman, M., & Lupu, E., "A Survey of Policy Specification Approaches", *Technical Report, Department of Computing, Imperial College of Science Technology and Medicine, London*, 2002.

[11] di Vimercati, S. D. C., Foresti, S., Jajodia, S., & Samarati, P., "Access Control Policies and Languages in Open Environments", *Secure Data Management in Decentralized Systems*, 2007, pp. 21-58.

[12] Do, H. H., Melnik, S., & Rahm, E., "Comparison of Schema Matching Evaluations", *Web, Web-Services, and Database Systems, LNCS* (2593), 2003, pp. 221-237.

[13] Dong, C., Russello, G., & Dulay, N., "Flexible Resolution of Authorisation Conflicts in Distributed Systems", *Proceedings of the 19th International Workshop on Distributed Systems: Operations and Management* (*DSOM*), 2008, pp. 95-108.

[14] Fatema, K. & Chadwick, D., "Resolving Policy Conflicts-Integrating Policies from Multiple Authors", *Proceedings of the International Conference on Advanced Information Systems Engineering* (*CAiSE*), 2014, pp. 310-321.

[15] Guo, Y., Pan, Z., & Heflin, J., "An Evaluation of Knowledge Base Systems for Large OWL Datasets", *Proceedings of the 3rd International Semantic Web Conference* (*ISWC*), 2004, pp. 274-288.

[16] Hu, H., Ahn, G., & Kulkarni, K., "Discovery and Resolution of Anomalies in Web Access Control Policies", *IEEE Transactions on Dependable and Secure Computing*, Vol. 10, No. 6, 2013, pp. 341-354.

[17] Jajodia, S., Samarati, P., Sapino, M. L., & Subrahmanian, V., "Flexible Support for Multiple Access Control Policies", *ACM Transactions on Database Systems* (*TODS*), Vol. 26, No. 2, 2001, pp. 214-260.

[18] Joshi, J., "Overview of Access Control Models", *IS 3350 Doctoral Seminar (Systems and Technology) Focus: Security and Privacy Assured Health Informatics*, 2015.

[19] Kamoda, H., Yamaoka, M., Matsuda, S., Broda, K., & Sloman, M., "Policy Conflict Analysis using Free Variable Tableaux for Access Control in Web Services Environments", *Proceedings of the 14th International World Wide Web Conference* (*WWW*), 2005, pp. 121-126.

[20] Kuang, T.P., Ibrahim, I., Sidi, F., & Udzir, N.I., "An Effective Modality Conflict Model for Identifying Applicable Policies during Policy Evaluation", *Journal of Advances in Computer Engineering and Technology (JACET)*, Vol. 4, No. 4, 2018, pp. 255-266.

[21] Lin, D., Rao, P., Ferrini, R., Bertino, E., & Lobo, J., "A Similarity Measure for Comparing XACML Policies", *IEEE Transactions on*

Knowledge and Data Engineering, Vol. 25, No. 9, 2013, pp. 1946-1959.

[22] Liu, A. X., Chen, F., Hwang, J., & Xie, T., "Designing Fast and Scalable XACML Policy Evaluation Engines", *IEEE Transactions on Computers*, Vol. 60, No. 12, 2011, pp. 1802-1817.

[23] Lupu, E. & Sloman, M., "Conflict Analysis for Management Policies", *Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management*, 1997, pp. 430-443.

[24] Martin, E., Xie, T., & Yu, T. "Defining and Measuring Policy Coverage in Testing Access Control Policies", *Proceedings of the 8th International Conference on Information and Communications Security* (*ICICS*), 2006, pp. 139-158.

[25] Moffett, J. D. & Sloman, M. S., "Policy Conflict Analysis in Distributed System Management", *Journal of Organizational Computing and Electronic Commerce*, Vol. 4, No. 1, 1994, pp. 1-22.

[26] Mohan, A. & Blough, D. M., "An Attribute-based Authorization Policy Framework with Dynamic Conflict Resolution", *Proceedings of the 9th Symposium on Identity and Trust on the Internet* (*IDTRUST*), 2010, pp. 37-50.

[27] Mohan, A., Blough, D. M., Kurc, T., Post, A., & Saltz, J., "Detection of Conflicts and Inconsistencies in Taxonomy Based Authorization Policies", *Proceedings of the 2011 IEEE International Conference on Bioinformatics and Biomedicine* (*BIBM*), 2011, pp. 590-594.

[28] Neri, M. A., Guarnieri, M., Magri, E., Mutti, S., & Paraboschi, S., "Conflict Detection in Security Policies using Semantic Web Technology", *Proceedings of the 1st AESS European Conference on Satellite Telecommunications* (*ESTEL*), 2012, pp. 1-6.

[29] Ngo, C., Demchenko, Y., & Laat, C. D., "Decision Diagrams for XACML Policy Evaluation and Management", *Journal of Computers and Security*, Vol. 49, 2015, pp. 1-16.

[30] Priebe, T., Dobmeier, W., Schläger, C., & Kamprath, N., "Supporting Attribute Based Access Control in Authorization and Authentication Infrastructures with Ontologies", *Journal of Software*, Vol. 2, No. 1, 2007, pp. 27-38.

[31] Proctor, S., "Sun's XACML implementation", 2004, URL: *http://sunxacml.sourceforge.net/*.

[32] Reul, Q. & Zhao, G., "Enabling Access to Web Resources through SecPODE-based Annotations", *Proceedings of the 2010 Confederated International Conferences on the Move to Meaningful Internet Systems* (*OTM*), 2010, pp. 596-605.

[33] Røstad, L., "*Access Control in Healthcare Information Systems*", PhD Thesis, Norwegian University of Science and Technology, Norway, 2008.

[34] Russello, G., Dong, C., & Dulay, N., "Authorisation and Conflict Resolution for Hierarchical Domains", *Proceedings of the Eighth IEEE International Workshop on Policies for Distributed Systems and Networks* (*POLICY*), 2007, pp. 201-210.

[35] Samarati, P. & di Vimercati, S. D. C., "Access Control: Policies, Models, and Mechanisms", *International School on Foundations of Security Analysis and Design*, *LNCS* (2171), 2001, pp. 137-196.

[36] Shaikh, R. A., Adi, K., & Logrippo, L., "A Data Classification Method for Inconsistency and Incompleteness Detection in Access Control Policy Sets", *International Journal of Information Security*, 2016, pp. 1-23.

[37] Shafiq, B., Joshi, J. B., Bertino, E., & Ghafoor, A., "Secure Interoperation in a Multidomain Environment Employing RBAC Policies", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 11, 2005, pp. 1557-1577.

[38] Singh, K. & Singh, S., "Design and Evaluation of XACML Conflict Policies Detection Mechanism", *International Journal of Computer Science and Information Technology*, Vol. 2, 2010, pp. 65-74.

[39] Stepien, B. & Felty, A., "Using Expert Systems to Statically Detect Dynamic Conflicts in XACML", *Proceedings of the 11th International Conference on Availability, Reliability and Security* (*ARES*), 2016.

[40] Stoller, S. D., Yang, P., Ramakrishnan, C. R., & Gofman, M. I., "Efficient Policy Analysis for Administrative Role Based Access Control", *Proceedings of the 14th ACM Conference on Computer and Communications Security* (*CCS*), 2007, pp. 445-455.

[41] Xia, X., "A Conflict Detection Approach for XACML Policies on Hierarchical Resources", *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications* (*GREENCOM*), 2012, pp. 755-760.