TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/10 pp478-488
Volume 22, Number 5, October 2017

Conflict Analysis and Detection Based on Model Checking for Spatial Access Control Policy

Aijuan Zhang, Cheng Ji*,Yu Bao, and Xin Li

Abstract: In this paper, we propose a Multi-granularity Spatial Access Control (MSAC) model, in which multi-granularity spatial objects introduce more types of policy rule conflicts than single-granularity objects do. To analyze and detect these conflicts, we first analyze the conflict types with respect to the relationship among the policy rules, and then formalize the conflicts by template matrices. We designed a model-checking algorithm to detect potential conflicts by establishing formalized matrices of the policy set. Lastly, we conducted experiments to verify the performance of the algorithm using various spatial data sets and rule sets. The results show that the algorithm can detect all the formalized conflicts. Moreover, the algorithm's efficiency is more influenced by the spatial object granularity than the size of the rule set.

Key words: spatial object; multi-granularity; conflict detection; model-checking

1 Introduction

In light of the economic value of geographic data and their importance to national security, spatial access control has become a hot issue in Geographic Information System (GIS) security research. For vector spatial data, space access control in GIS services is required to control users' access based on the multiple granularities of geographic layers and geographic features[1]. There have been a number of studies focused on spatial access control models. Sasaoka and Medeiros[2] proposed a spatial data authorization model for spatial databases. Bertino et al.[3] extended the Role-Based Access Control (RBAC) model to the GEO-RBAC that can handle the access control to the spatial and location-based information based on the mobile user's physical and logical locations, and then the access control models with physical location and time constraints were put forward[4–6]. In distributed GIS, spatial access control models[7–10] were proposed to implement cross-system authorization, but these models did not consider the spatial data characteristics and could not meet the requirements of multi-granularity spatial data control.

To ensure the validity of models, potential conflicts must also be detected. The main conflict detection methods are the formal method and the test-based authentication method. The former method has two types: model checking and theorem proving. The Multi-Terminal Binary Decision Diagrams (MTBDD)[11] and Perti nets[12] were used to formalize access control policy. However, the conflict types are different in various systems. To detect the inconsistent policies, the various precedence relationships established between policies were discussed[13], and then SVM was trained to discover attack patterns[14]; matrix was used to formalize conflict pattern[15]. In addition, resource semantic tree and state relativity were used to detect the conflicts in XACML policy[16]. In summary, the resources and models for each system differ.

To realize the access control to spatial objects in a service-oriented spatial data infrastructure, we propose the Multi-granularity Spatial Access Control (MSAC)

• Aijuan Zhang, Cheng Ji, and Yu Bao are with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China. E-mail: {zaj, jicheng, baoyu}@cumt.edu.cn.

• Xin Li is with the School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China. E-mail: linuxcumt@126.com.

∗ To whom correspondence should be addressed.

Manuscript received: 2016-10-21; accepted: 2016-12-21

model[17], which adds subject's location and object constraints. Based on the spatial, semantic, and feature-field constraints, the multi-granularity object access control can be realized. Since MSAC extends the spatial attributes and constraints, the conflict types must be redefined and a new conflict detection algorithm designed.

The rest of this paper is organized as follows. In Section 2, we outline MSAC model and its control effect in GIS, and in Section 3, we conduct the policy conflict analysis. In Section 4, we detail the procedure for formalizing the rule system and conflict patterns, and then we design the conflict detection algorithm based on model checking. In Section 5, we present our experimental results, and in Section 6, we draw our conclusions.

2 MSAC Model

In this section, we propose the MSAC model that extends the core RBAC to include attributed constraints. In addition to the spatial data attributes, the subject's location is also included in the control model. There are three types of authorization granularity: geographic layer, feature object, and feature object view. Various constraints can express different control granularity: scale and time constraints can control graphic layer granularity; topological and semantic constraints can control feature object granularity; and field constraints can control feature object view granularity. The MSAC is shown in Fig. 1.

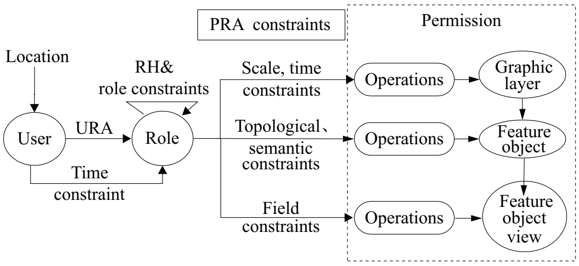


Fig. 1 MSAC model.

In this model, subjects add the location attribute. Scale, mapping time, topological, semantic, and field constraints arise during the Permission-Role Assignment (PRA). In the following, we introduce each element of the model.

- **User**=(identifier, location, roles), which visits the system. We use the identifier attribute to confirm the user's identity, and the location attribute, which is a logic or real region or point, to express topological constraints.
- **Role**=(name, PS), the permission assignment unit. When a user establishes a session, the roles of the user are active and their assigned location attributes are the same as the user's. The user is assigned to the Permission Set (PS) of the roles, and the role constraints implement Role Hierarchy (RH) and duty separation.
- **Operations**: methods used by a spatial class.
- **Targets**: spatail objects which includes the uppermost spatial objects: graphic layer; the second-layer spatial objects: feature objects; and the lowest-layer spatial objects: feature object views. These objects form a hierarchical structure.
- **URA**$\subseteq$(User$\times$Role)$_{\mathrm{C1}}$, in condition C1, roles are assigned to a user and the user's location is assigned to the roles.
- **C1**: the URA context time constraint.
- **PRA**$\subseteq$(permission$\times$role)$_{\mathrm{C2}}$, mapping from the role set to the permission set in condition C2.
- **C2**: PRA constraints$\subset$\{topological, scale, mapping time, semantic, and field constraints\}, used for multi-granularity spatial object control.
- **RH**$\subseteq$(Role$\times$Role), the partial ordering relation of the role sets, which represents the inheritance relationship between two roles. If (role$_1$, role$_2$) $\in$RH and the relationship is defined as role$_2\lhd$role$_1$, and then it indicates that role$_2$ extends the permission of role$_1$.

3 Policy Conflict Analysis

3.1 Description of security policy

In this section, we present MSAC model by defining the policy rules. The policy rule construct can be represented in a quintuple form: $\langle$Subject, Operation, Target, Condition, Effect$\rangle$, and we use five abbreviated characters to describe these five elements: S, O, T, C, and E.

- Subject: the role, or user to whom the rule applies.
- Target: geographic data set.
- Condition: constraints in which permission is effective, and it can also be considered as a spatial query expression.
- Effect= permit, deny.

Figure 2 shows the results of the multi-granularity access control module.

When a request is sent to a service interface, the access is controlled by the MSAC system. We

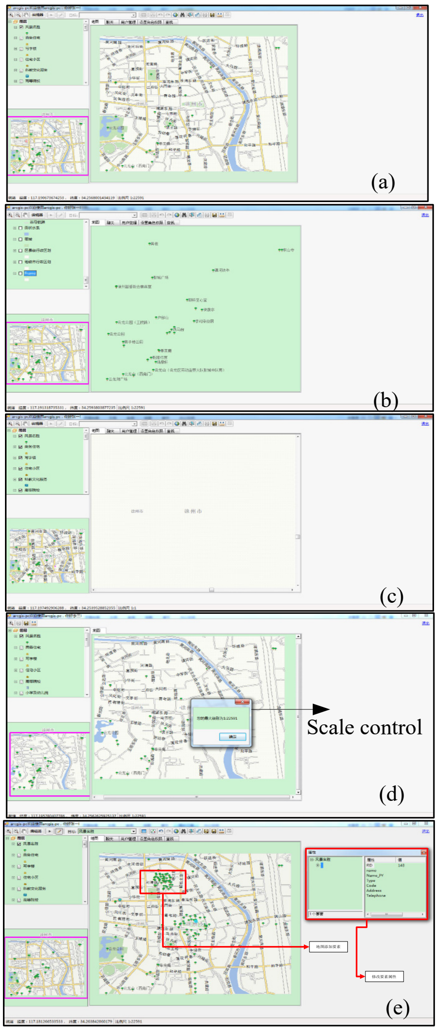


Fig. 2 Multi-granularity object control: complete positive authorization (a); layer authorization (b); feature object authorization (c); layer authorization with scale constraints (d); field view object authorization (e).

developed this system using the Arcgis Engine. Figure 2a shows a complete visit to the map, which is composed of forty layers; Fig. 2b shows only a partial map using graphic-layer control. Figure 2c shows part of the features in a special graphic layer; and Fig. 2d shows the layers with a 22 591-feature scale constraint. If the map was amplified beyond this scale, the map service would not provide more detailed features. Some spatial objects' part information is shown in Fig. 2e.

There are three access control rules, by which a user "John", for example, accesses "city" with different control granularities, and the fine-granularity control can be achieved by adding constraints:

⟨ John; city; null; null; getFeatureInfo; permit ⟩,
⟨ John; city; scale= {1:1 1:25}; getFeatureInfo; deny ⟩,
⟨ John; city; **within** "10.1 20.1 10.6 24.2 13.3 26.1", **area**<=50000; name, area, description; getFeatureInfo; permit ⟩.

The bold words above are spatial constraint functions and semantic attribute constraint expressions, respectively.

3.2 Policy rule analysis

Assuming there are two rules: R_1 and R_2, Table 1 describes the relationship of the counterpart elements, denoted as $\Re$.

We use the "effect" element in an access control rule to present positive or negative permission. If the values of the "effect" elements are contradictory for two rules, and the rules are performed on related subjects, targets, and operations, then pattern conflicts may arise. If the values of the "effect" elements are the same and subjects/targets are mutually exclusive, there may be duty-separation conflicts. In addition, rules can have redundant conflicts, which include the same "effect" element and intersect with other elements. These rules only increase the storage capacity of the rules.

There are collection relations ("sub" represents a true subset, "sup" represents a true superset, "equ" represents equality, "cor" represents intersection), $\lhd$ and $\rhd$ represent hierarchical relations, as well as

Table 1 Relationship between counterpart elements (A^+= permit, A^-= deny, N= any related element).

Relationship: $\Re$		Description
Effort	ctd1: $R_1.E\Theta\ R_2.E$	$(R_1.E = \mathrm{A}^+)\wedge$ $(R_2.E = \mathrm{A}^-)$
	ctd2: $R_1.E\Theta\ R_2.E$	$(R_1.E = \mathrm{A}^-)\wedge$ $(R_2.E = \mathrm{A}^+)$
	equ	$((R_1.E = \mathrm{A}^+)\wedge$ $(R_2.E = \mathrm{A}^+)\vee$ $(R_1.E = \mathrm{A}^-)\wedge$ $(R_2.E = \mathrm{A}^-))$
Subject target operation condition	sub	$R_1.N \subset R_2.N$
	sup	$R_1.N \supset R_2.N$
	equ	$R_1.N = R_2.N$
	cor $R_1.N \sim R_2.N$	$(R_1.N \cap R_2.N \neq \text{null})$ $\wedge(R_1.N \not\subseteq R_2.N)\wedge$ $(R_2.N \not\subseteq R_1.N)$
	unr	$R_1.N \cap R_2.N = \text{null}$
Subject target	mux: $R_1.N \neq R_2.N$	$\neg(R_1.N \wedge R_2.N)$
	$R_1.N \lhd R_2.N$	$R_2.N$ is upper object $R_1.N$ is lower object
	$R_1.N \rhd R_2.N$	$R_2.N$ is lower object $R_1.N$ is upper object

semantic relations such as “mux”, which indicates mutual exclusion. The upper resource is a coarse-grained target object, whereas the lower resource is a fine-grained target object.

In this work, we focus on permission conflicts. There are two types of permission conflicts: redundant and pattern conflicts. Although rules with redundant conflicts will only increase storage capacity, rules with pattern conflicts may lead to contradictory judgments. Therefore, here, we focus on detecting pattern conflicts.

There are both hierarchical and intersecting relations between subjects/targets, which forms permission inheritance and contain, and causes conflicts. Within the subject hierarchy relation, the lower subject has the permissions of the upper subject, whereas in the target objects hierarchy relation, the coarse-grained upper object will hand over “deny” permission to fine-grained lower objects. In the following section, we propose a new conflict classification method with respect to the target relationship.

When analyzing possible conflicts, we must compare the rules states. Here we assume there to be two rules: R_1 and R_2, and the respective rule states can be expressed as State_1 and State_2.

$$\text{State}_1 = (\bigcup_{n=1}^{n1} S_{1,n}, \bigcup_{n=1}^{n2} T_{1,n}, \bigcup_{n=1}^{n3} O_{1,n}, \bigcup_{n=1}^{n4} C_{1,n}),$$

$$\text{State}_2 = (\bigcup_{m=1}^{m1} S_{2,m}, \bigcup_{m=1}^{m2} T_{2,m}, \bigcup_{m=1}^{m3} O_{2,m}, \bigcup_{m=1}^{m4} C_{2,m}).$$

According to the relationship between the targets, there are different relationship types between State_1 and State_2, which include target hierarchy, target coverage and intersection, and target independence.

Target hierarchy: If the following relations exist between State_1 and State_2, there will be a target hierarchy relation between them.

$$((T_{1,n} \lhd T_{2,m}) \wedge (\bigcup_{n=1}^{n1} S_{1,n} \cap \bigcup_{m=1}^{m1} S_{2,m} \neq \varnothing)) \vee$$
$$((T_{1,n} \rhd T_{2,m}) \wedge (\bigcup_{n=1}^{n1} S_{1,n} \cap \bigcup_{m=1}^{m1} S_{2,m} \neq \varnothing)) \wedge$$
$$(\bigcup_{n=1}^{n3} O_{1,n} \cap \bigcup_{m=1}^{m3} O_{2,m} \neq \varnothing) \wedge (\bigcup_{n=1}^{n4} C_{1,n} \cap$$
$$\bigcup_{m=1}^{m4} C_{2,m} \neq \varnothing).$$

The targets have a “is-a” relationship and the subjects are related, as are the operations/conditions.

Target coverage and intersection: If the following relations exist between State_1 and State_2, the target of State_1 is covered by or intersects with the target of State_2.

$$((\forall T_{1,n}, \exists T_{2,m} : T_{1,n} \in \bigcup_{m=1}^{m2} T_{2,m}) \vee$$
$$(\bigcup_{n=1}^{n2} T_{1,n} \cap \bigcup_{m=1}^{m2} T_{2,m} \neq \varnothing)) \wedge$$
$$(\bigcup_{n=1}^{n1} S_{1,n} \cap \bigcup_{m=1}^{m1} S_{2,m} \neq \varnothing) \wedge$$
$$(\bigcup_{n=1}^{n3} O_{1,n} \cap \bigcup_{m=1}^{m3} O_{2,m} \neq \varnothing) \wedge$$
$$(\bigcup_{n=1}^{n4} C_{1,n} \cap \bigcup_{m=1}^{m4} C_{2,m} \neq \varnothing).$$

These targets have a containing or intersecting relationship, as do the operations and condition attributes of the two rules.

Target independence: If the following relations exist between State_1 and State_2, State_1 is irrelevant with State_2.

$$(\forall T_{1,n} : T_{1,n} \notin \bigcup_{m=1}^{m2} T_{2,m}) \wedge$$
$$((T_{1,n} \neg \lhd T_{2,m}) \wedge (T_{1,n} \neg \rhd T_{2,m})).$$

If the above two relations exist between the rules, a conflict may exist. Within a subject hierarchy relation, the lower subject will have the permissions of the upper subject; whereas in a target objects hierarchy relation, the upper object will hand over “deny” permission to fine-grained lower objects. In the following section, we propose a new conflict classification method specific only to the target object relation.

3.3 Conflicts between hierarchical targets

We assume the targets to have hierarchical relations and the same operations but different efforts in the two rules. Figure 3 shows all the states between these two rules.

Hierarchical objects will hand over “deny” permissions to lower objects, and never hand over “permit” permission to lower objects. We assume a subject S and a resource object T, if S_1 is the upper subject of S_2, and T_1 is the upper resource of T_2, the inheritance rules can be expressed as follows:

$$(S_1, T, C) \rightarrow \text{permit} \Rightarrow (S_2, T, C) \rightarrow \text{permit},$$
$$(S_1, T, C) \rightarrow \text{deny} \Rightarrow (S_2, T, C) \rightarrow \text{deny}.$$

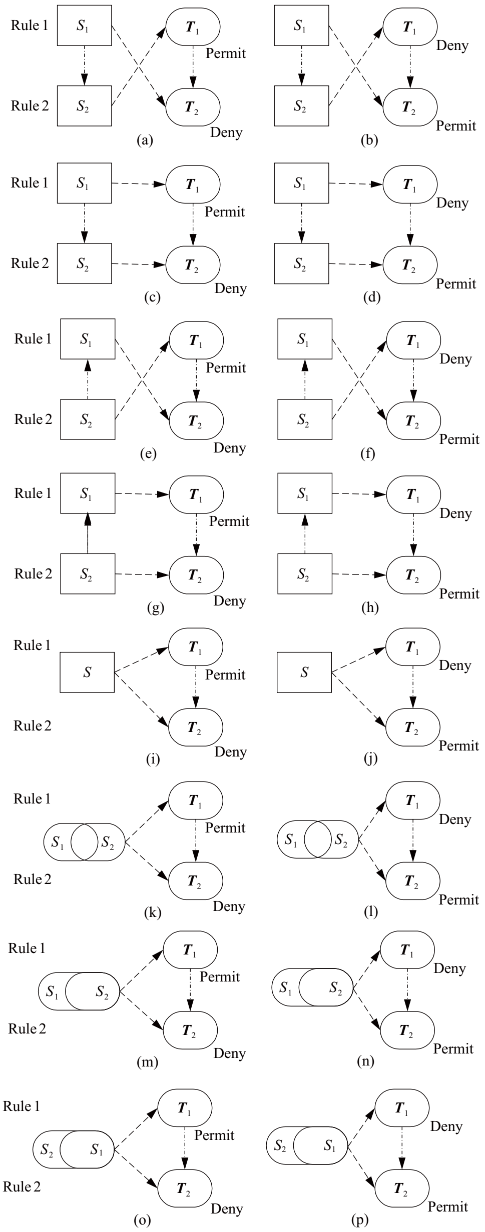


Fig. 3 Rule states of hierarchical targets.

The object hierarchy relation can lead to the following permission rules.

$$(S, T_1, C) \to \text{permit} \nRightarrow (S, T_2, C) \to \text{permit},$$

$$(S, T_1, C) \to \text{deny} \Rightarrow (S, T_2, C) \to \text{deny}.$$

According to the above properties, we can deduce the following sixteen situations.

(1) $\text{State}_1 = (S_1, T_2, O, C)^{\text{deny}}$,

$\text{State}_2 = (S_2, T_1, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_2, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{deny}} \\ \\ \text{State}_2 = (S_2, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \end{array}\right\} \nRightarrow \text{cf.}$$

(2) $\text{State}_1 = (S_1, T_2, O, C)^{\text{permit}}$,

$\text{State}_2 = (S_2, T_1, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_2, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \Rightarrow \text{cf.}$$

(3) $\text{State}_1 = (S_1, T_1, O, C)^{\text{permit}}$,

$\text{State}_2 = (S_2, T_2, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(4) $\text{State}_1 = (S_1, T_1, O, C)^{\text{deny}}$,

$\text{State}_2 = (S_2, T_2, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{deny}} \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \end{array}\right\} \Rightarrow \text{cf.}$$

(5)$\text{State}_1 = (S_1, T_2, O, C)^{\text{deny}}$,

$\text{State}_2 = (S_2, T_1, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_2, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{deny}} \\ \\ \text{State}_2 = (S_2, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(6) $\text{State}_1 = (S_1, T_2, O, C)^{\text{permit}}$,

$\text{State}_2 = (S_2, T_1, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_2, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \Rightarrow \text{cf.}$$

(7) $\text{State}_1 = (S_1, T_1, O, C)^{\text{permit}}$,

$\text{State}_2 = (S_2, T_2, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(8) $\text{State}_1 = (S_1, T_1, O, C)^{\text{deny}}$,
$\text{State}_2 = (S_2, T_2, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{deny}} \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \end{array}\right\} \Rightarrow \text{cf.}$$

(9) $\text{State}_1 = (S, T_1, O, C)^{\text{permit}}$,
$\text{State}_2 = (S, T_2, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_1 = (S, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(10) $\text{State}_1 = (S, T_1, O, C)^{\text{deny}}$,
$\text{State}_2 = (S, T_2, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S, T_2, O, C)^{\text{deny}} \\ \\ \text{State}_2 = (S, T_2, O, C)^{\text{permit}} \end{array}\right\} \Rightarrow \text{cf.}$$

(11) $\text{State}_1 = (S_1, T_1, O, C)^{\text{permit}}$,
$\text{State}_2 = (\mathrm{S_2, T_2, O, C})^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(12) $\text{State}_1 = (S_1, T_1, O, C)^{\text{deny}}$,
$\text{State}_2 = (S_2, T_2, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{deny}}(\text{part}) \\ \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{deny}}(\text{part}) \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \end{array}\right\} \Rightarrow \text{cf (part).}$$

(13) $\text{State}_1 = (S_1, T_1, O, C)^{\text{permit}}$,
$\text{State}_2 = (S_2, T_2, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{permit}} \nRightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(14) $\text{State}_1 = (S_1, T_2, O, C)^{\text{permit}}$,
$\text{State}_2 = (S_2, T_1, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_2 = (S_1, T_2, O, C)^{\text{deny}} \end{array}\right\} \Rightarrow \text{cf.}$$

(15) $\text{State}_1 = (S_1, T_1, O, C)^{\text{permit}}$,
$\text{State}_2 = (S_2, T_2, O, C)^{\text{deny}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{permit}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{permit}}(\text{part}) \nRightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{permit}} \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{deny}} \end{array}\right\} \nRightarrow \text{cf.}$$

(16) $\text{State}_1 = (S_1, T_1, O, C)^{\text{deny}}$,
$\text{State}_2 = (S_2, T_2, O, C)^{\text{permit}}$,

$$\left\{\begin{array}{l} \text{State}_1 = (S_1, T_1, O, C)^{\text{deny}} \Rightarrow \\ \text{State}_1 = (S_2, T_1, O, C)^{\text{deny}}(\text{part}) \Rightarrow \\ \text{State}_1 = (S_2, T_2, O, C)^{\text{deny}}(\text{part}) \\ \\ \text{State}_2 = (S_2, T_2, O, C)^{\text{permit}} \end{array}\right\} \Rightarrow \text{cf (part).}$$

The "cf" in the above equations represents "conflict". Then we can conclud that conflicts may appear in these situations: (2), (4), (6), (8), (10), (12), (14), and (16).

We can formalize these target hierarchy conflicts as follows: If there is a conflict of this type between rule R_1 and R_2, it must meet the following conditions:

$$\begin{aligned} &((R_1.T \lhd R_2.T) \wedge (R_1.E = A^+) \wedge \\ &(R_2.E = A^-)) \wedge (R_1.S\Re_x R_2.S) \wedge \\ &(R_1.O\Re_z R_2.O) \wedge (R_1.C\Re_m R_2.C), \\ &\Re_x \in \{\subset, \supset, =, \sim, \lhd, \rhd\}, \Re_z, \Re_m \in \{\subset, \supset, =, \sim\} \end{aligned} \tag{1}$$

That is to say, with the upper object granted the deny right and the lower object granted the permit right, these two rules will conflict when the conditions and actions intersect respectively, irrespective of whether the relationship between the subject elements is equal, containing, contained, hierarchical or intersecting. $\Re_x$ represents the relation between subjects, $\Re_z$ represents the relation between operations, and $\Re_m$ between conditions.

3.4 Conflicts of intersecting, equal, and covering targets

We assume the targets of two rules do not have hierarchical relations, then the relation between them, excluding independence, may be either intersecting, coverage, or equality. If the subjects are associated with each other, between which the relation may be intersection, coverage, containment, equality or hierarchy, and the "effort" is contrary. Figure 4 shows the possible states, and all of which conflict.

The above conflicts are formalized. If there is a conflict of this type between rule R_1 and R_2, it must meet the following conditions:

$$\begin{aligned}&((R_1.E = A^+) \wedge (R_2.E = A^-)\vee\\&(R_1.E = A^-) \wedge (R_2.E = A^+))\wedge\\&(R_1.S\Re_x R_2.S) \wedge (R_1.T\Re_y R_2.T)\wedge\\&(R_1.O\Re_z R_2.O)(R_1.C\Re_m R_2.C),\\&\Re_x \in \{\subset, \supset, =, \sim, \lhd, \rhd\},\\&\Re_y, \Re_z, \Re_m \in \{\subset, \supset, =, \sim\}\end{aligned} \tag{2}$$

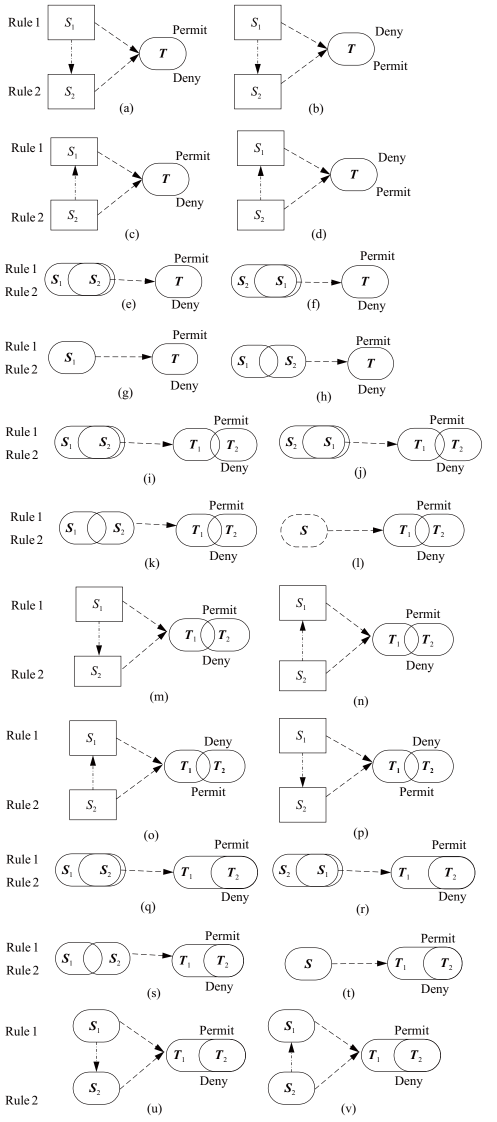


Fig. 4 Rule states of hierarchical targets.

$\Re_y$ represents the relation between targets.

4 Conflict Detection Algorithm Based on Matrix Model-Checking

4.1 Matrix model of the rule system

In this paper, we use model-checking method to detect pattern conflicts. Bonatti et al.[18] proposed the classic policy synthesis algebraic model, the basic idea of which is to define the access control policy by triples collection. Here, we break down the policy rules into a collection[19,20], i.e., <subject, target, operation, condition, effect>, and use a matrix to describe the relationships between the corresponding elements in any two rules. To create these five relation matrices, we establish a Subject-Directed Graph (SDG) and a rule table: ruleTab. We introduce the algorithm in the following.

Assuming there are n rules, then we can formalize the rule system by five $n\times n$ matrices in which each element represents its relation of the corresponding elements in two rules, which we describe as follows.

$$\boldsymbol{M_S} = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & \cdots & \varphi_{1n}\\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \cdots & \varphi_{2n}\\ \varphi_{31} & \varphi_{32} & \varphi_{33} & \cdots & \varphi_{3n}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \varphi_{n1} & \varphi_{n2} & \varphi_{n3} & \cdots & \varphi_{nn} \end{pmatrix},$$

$$\boldsymbol{M_T} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \cdots & \alpha_{2n}\\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \cdots & \alpha_{3n}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \alpha_{n1} & \alpha_{n2} & \alpha_{n3} & \cdots & \alpha_{nn} \end{pmatrix},$$

$$\boldsymbol{M_O} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \cdots & \theta_{1n}\\ \theta_{21} & \theta_{22} & \theta_{23} & \cdots & \theta_{2n}\\ \theta_{31} & \theta_{32} & \theta_{33} & \cdots & \theta_{3n}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \theta_{n1} & \theta_{n2} & \theta_{n3} & \cdots & \theta_{nn} \end{pmatrix},$$

$$\boldsymbol{M_C} = \begin{pmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \cdots & \beta_{1n}\\ \beta_{21} & \beta_{22} & \beta_{23} & \cdots & \beta_{2n}\\ \beta_{31} & \beta_{32} & \beta_{33} & \cdots & \beta_{3n}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \beta_{n1} & \beta_{n2} & \beta_{n3} & \cdots & \beta_{nn} \end{pmatrix},$$

$$\boldsymbol{M_E} = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \cdots & \omega_{1n}\\ \omega_{21} & \omega_{22} & \omega_{23} & \cdots & \omega_{2n}\\ \omega_{31} & \omega_{32} & \omega_{33} & \cdots & \omega_{3n}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \omega_{n1} & \omega_{n2} & \omega_{n3} & \cdots & \omega_{nn} \end{pmatrix}.$$

We express the value of any matrix element by seven bits that represent the possible relation between two elements. These seven bits are as follows: sub, sup, equ, cor, mux, $\lhd$, and $\rhd$. Not all of the relationships exist between the counterpart elements. Therefore, if a certain relation does not exist, we use "null" to express the bit value. Each element of the above five matrices can be used to express the following relation vectors:

$\boldsymbol{M_S}[i,j] = \boldsymbol{\varphi}_{ij} = \{\text{sub}, \text{sup}, \text{equ}, \text{cor}, \text{mux}, \lhd, \rhd\}$,

$\boldsymbol{M_T}[i,j] = \boldsymbol{\alpha}_{ij} = \{\text{sub}, \text{sup}, \text{equ}, \text{cor}, \text{null}, \lhd, \rhd\}$,

$M_O[i,j] = \boldsymbol{\theta}_{ij} = \{\text{sub}, \text{sup}, \text{equ}, \text{cor}, \text{null}, \text{null}, \text{null}\}$,

$\boldsymbol{M_C}[i,j] = \boldsymbol{\beta}_{ij} = \{\text{sub}, \text{sup}, \text{equ}, \text{cor}, \text{null}, \text{null}, \text{null}\}$,

$\boldsymbol{M_E}[i,j] = \boldsymbol{\omega}_{ij} = \{\text{ctd}_1, \text{ctd}_2, \text{equ}, \text{null}, \text{null}, \text{null}, \text{null}\}$.

To obtain these five matrices, we design a syntax parser to parse the rules into five separate elements, and the ruleTab and SDG can be obtained in the process, at the same time, we can obtain the resource object tree, OBTree, using granularity constraints. Based on these five matrices, we can also detect the duty separation conflict.

4.2 Matrix model of conflicts

To determine whether a conflict exists between two rules, we must obtain a relationship vector from the five matrices, which can be described as follows:

$$(\boldsymbol{\varphi}_{i,j}, \boldsymbol{\alpha}_{i,j}, \boldsymbol{\theta}_{i,j}, \boldsymbol{\beta}_{i,j}, \boldsymbol{\omega}_{i,j})^{\mathrm{T}} \tag{3}$$

We can use this state vector to express the conflicts analyzed in Section 2.

- According to Formula (1), we can express the conflicts of hierarchical targets as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

- According to Formula (2), we can express the conflicts of the intersecting, equal, and covering targets as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

4.3 Pattern conflict detection algorithm

Algorithm 1 is described in Fig. 5.

Algorithm 1 Pattern conflict detection

Input: P // original policy set.

Output: conSet //conflict rules pair.
SDG //subject directed graph.
OBTree // resource object tree.
$\boldsymbol{M_S}, \boldsymbol{M_T}, \mathbf{M_O}, \boldsymbol{M_C}, \boldsymbol{M_E}$ //five $n \times n$ matrices
ruleTab //rule Table

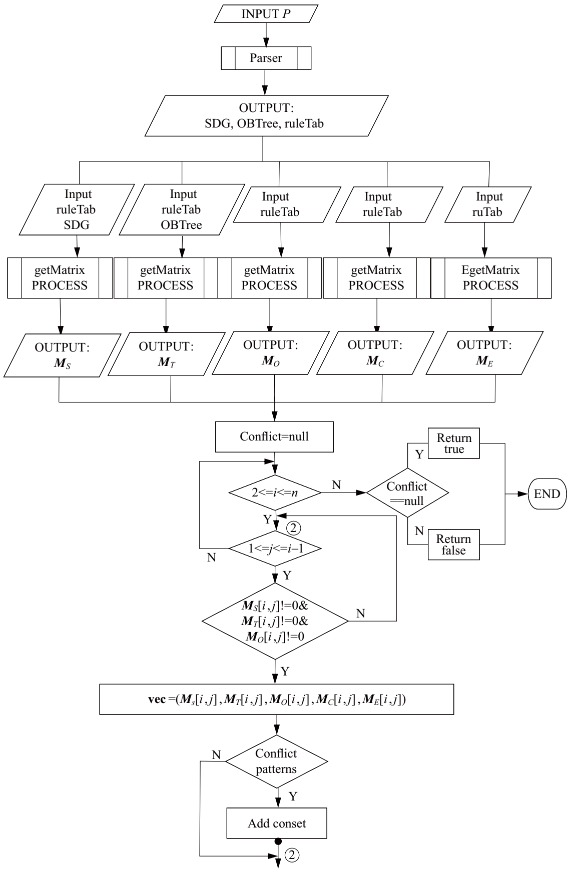


Fig. 5 Flowchart of the detection algorithm.

(1) Enter policy rule set P, then use the Parser module to obtain the SDG and rule table, ruleTab.

(2) Using the getMatrix() (shown in Fig. 6) and the EgetMatrix() (shown in Fig. 7), the algorithm obtains five $n \times n$ relationship matrices.

(3) If the rule pair, R_i and R_j, is relevant (that is, if the subjects of the two rules as well as the objects and operations are associated), a relationship vector, **vec**, is obtained from the five matrices, which can be compared with the conflict template in Formulas (4) and (5).

(4) Repeat Steps (2) and (3) until every pair of rules is tested.

(5) Return conSet.

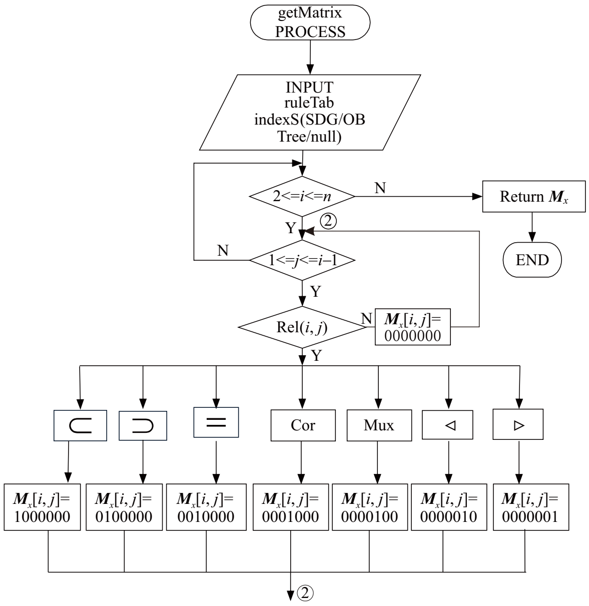


Fig. 6 Flowchart of creating matrices algorithm ($\boldsymbol{M_S}$, $\boldsymbol{M_T}$, $\boldsymbol{M_O}$, $\boldsymbol{M_C}$).

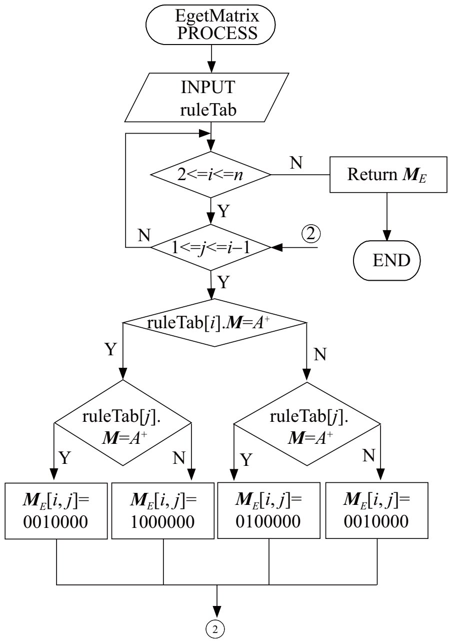


Fig. 7 Flowchart of the detection algorithm ($\boldsymbol{M_E}$).

We use the function of the sub-module: getMetrics to establish the four relationship matrices: $\boldsymbol{M_S}$, $\boldsymbol{M_T}$, $\boldsymbol{M_O}$, and $\boldsymbol{M_C}$, and use the function of the sub-module, EgetMetrix, to establish the $\boldsymbol{M_E}$ matrix only. In Fig. 6, $\boldsymbol{M_x}$ represents $\boldsymbol{M_S}$, $\boldsymbol{M_T}$, $\boldsymbol{M_O}$, and $\boldsymbol{M_C}$. This algorithm is described as follows:

(1) With the ruleTab, OBTree, and SDG created in the parser process, the Rel(int, int) module uses set operations and graph traversal operation to determine the element relationship between any two rules (shown in Fig. 6).

(2) To create the $\boldsymbol{M_E}$ matrix, we use the EgetMatrix module (shown in Fig. 7) to judge the relationship between the two "Effect" elements by the judgment operation.

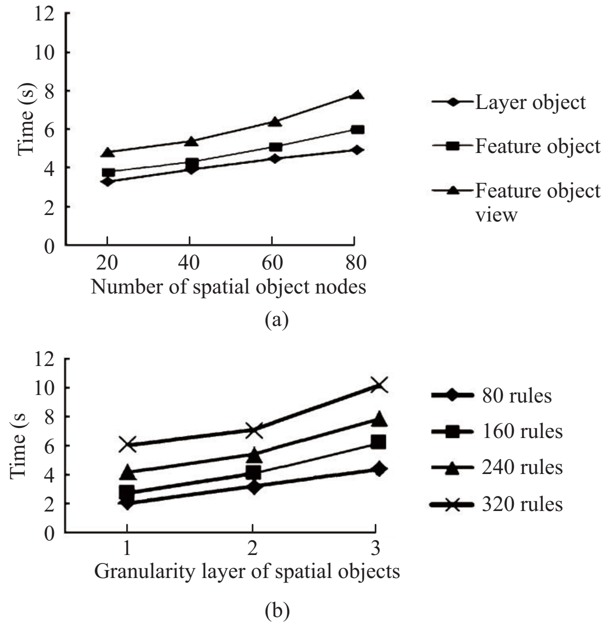
(3) Repeat Steps (1) and (2) until a matrix is created.

5 Results and Discussion

We use the China map with the scale 1:1 000 000 as the controlled target, and this map has nine layers: res1_4m, res2_4m, diqujie_polyline, hyd2_4l, rai_4m, row_4m, bou2_4l, hyd2_4p, and bou2_4p. We set up access control rules with three granularities for four classes of targets. The targets in the graphic layer granularity are with scale and time constraints; the feature targets are objects with semantic and topological constraints and the feature object views with field constraints. If objects have topological constraints, we must confirm the topological relation between the two geometries.

In this experiment, we used an original policy set as input to detect possible conflict pairs. In the first experiment, we analyzed how the influence of the number of spatial-object nodes on the algorithm efficiency as shown in Fig. 8.

There were four resource trees and the number of nodes was 20, 40, 60, and 80, respectively. The spatial data and the same number of policy rules were evenly distributed in each layer. With an increase in the number



(a)

(b)

Fig. 8 Efficiency as a function of the number of spatial object nodes (a) and efficiency in different layers (b).

of data nodes, the detection time also increased, as shown in Fig. 8a.

The results of the first experiment shows that if the policy rules are deployed in one level, the number of spatial objects has almost the same influence for each granularity.

In the second experiment, for the same number of spatial data nodes, rule conflicts are detected in different resource layers. There are 40 nodes and the rule numbers of the four-group policy are 80, 160, 240, and 320, respectively. The experimental data show a steep increase in the detection time curve when we detect conflicts in multilayers as shown in Fig. 8b. Moreover, if the rules are deployed in the feature-view granularity, the running time increases more rapidly. Also, the trends associated with different numbers of rules differ only slightly, with the number of rules affecting the algorithm efficiency only slightly.

6 Conclusion

With respect to the relationship of counterpart elements between spatial objects in MSAC, we analyzed the types of conflicts in rules generated by the MSAC model. To detect conflicts in the rule set, we used a model-checking approach to design a conflict-detection algorithm, in which we formalized the relationships of counterpart elements between any two rules using five matrices. We then described these formalized conflicts in template matrices. We used a matching operation to generate the conflict detection algorithm, which shows that both the number of the rules and the granularities of spatial objects influence algorithm efficiency: the number of the rules influences only the formation process of the rule systems, whereas the levels of object granularities influence the trend of running time.

In this paper, we considered only those conflicts arising when the effects between two rules were opposing. If the effects are the same (permit or deny), there may be sequence and redundancy conflicts, which are easy to digest. On the other hand, in the URA process, due to the role hierarchies, there may also be role conflicts. Therefore, an additional algorithm is needed to detect these conflicts.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Nos. 51204185 and 41674030), Natural Youth Science Foundation of Jiangsu Province, China (No. BK20140185), China Postdoctoral Science Foundation (No. 2016M601909), and the Fundamental Research Funds for the Central Universities (No. 2014QNA44). We are grateful to the reviewers for their suggestions on improving this paper.

References

[1] A. Matheus, Declaration and enforcement of fine-grained access restrictions for a service-based geospatial data infrastructure, presented at the 10th ACM Symposium on Access Control MODELS and Technologies, Stockholm, Sweden, 2005.

[2] L. K. Sasaoka and C. B. Medeiros, Access control in geographic databases, in *Advances in Conceptual Modeling—Theory and Practice*. Springer Berlin Heidelberg, 2006, pp. 110–119.

[3] E. Bertino, B. Catania, M. L. Damiani, and P Perlasca, GEO-RBAC: A spatially aware RBAC, presented at 10th ACM Symposium on Access Control MODELS and Technologies, Stockholm, Sweden, 2005.

[4] S. M. Chandran and J. B. D. Joshi, LoT-RBAC: A location and time-based RBAC model, in *Web Information Systems Engineering-WISE 2005*, 2005, pp. 361–375.

[5] V. Atluri and S. A. Chun, A geotemporal role-based authorization system, *International Journal of Information and Computer Security*, vol. 1, no. 12, pp. 143–168, 2007.

[6] I. Ray and M. Toahehoodee, A spatio-temporal role-based access control model, in *Ifip Wg 11.3 Working Conf. on Data and Applications Security*, Springer-Verlag, 2007, pp. 211–226.

[7] M. L. Damiani, E. Bertino, and C. Silvestri, Spatial domains for the administration of location-based access control policies, *Journal of Network and Systems Management*, vol. 16, no. 3, pp. 277–302, 2008.

[8] A. Matheus, Security considerations on processing of geospatial information in the cloud, in 4th *Int. Conf. on Computing for Geospatial Research and Application*, 2013, pp. 82–86.

[9] G. Lin and D. Wang, MTBAC: A mutual trust based access control model in cloud computing, *China Communications*, vol. 11, no. 4, pp. 154–162, 2014.

[10] X. Ye, Privacy preserving and delegated access centrol for cloud application, *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 40–54, 2016.

[11] K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz, Verification and change-impact analysis of access control policies, in *Proc. of the 27th Int. Conference on Software Engineering*, St Louis, MO, USA, 2005, pp. 196–205.

[12] K. Knorr, Multilevel security and information flow in Petri net workflows, in *Proc. of the 9th International Conference on Telecommunication Systems-Modeling and Analysis*, 2001, pp. 9–20.

[13] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, Distributed denial of service attacks detection using support vector machine, *Information—An International Interdisciplinary Journal*, vol. 14, no. 1, pp. 127–134, 2011.

[14] S. Davy, B. Jennings, and J. Strassner, The policy continum-Policy authoring and conflict analysis, *Computer Communications*, vol. no. 31, pp. 2981–2995, 2008.

[15] C. L. Lee, G. Y. Lin, and Y. C. Chen, An efficient conflict detection algorithm for packet filters, *IEICE Trans. on Inf.& Sys.*, vol. 95, no. 2, pp. 472–479, 2012.

[16] Y. Z. Wang and D. G. Feng, A conflict and redundancy analysis method for XACML rules, (in Chinese), *Journal of Computers*, vol. 32, no. 3, pp. 516–529, 2009.

[17] A. J. Zhang, J. X. Gao, C. Ji, J. Sun, and Y. Bao, Multi-granularity spatial-temporal access control model for web GIS, *Trans. of Nonferrous Metals Society of China*, vol. 24, no. 9, pp. 2946–2953, 2014.

[18] P. Bonatti, S. D. Vimercad, and P. Samarali, An algebra for composing access control policies, *ACM Trans. on Inf. & Sys. Security*, vol. 5, no. 1, pp. 1–35, 2002.

[19] A. K. Bandara, A formal approach to analysis and refinement of policies, Ph.D. dissertation, Imperial College London, London, UK, 2005.

[20] H. Hamed, E. Al-Shaer, and W. Marrero, Modeling and verification of IPSec and VPN security policy, in *Proc. of the 13th IEEE International Conference on Network Protocols*, Boston, MA, USA, 2005, pp. 259–278.

Aijuan Zhang is currently an associated professor at the School of Computer Science and Technology, China University of Mining and Technology. She received the PhD, MS, and BS degrees from China University of Mining and Technology in 2012, 2005, and 2002, respectively. Her research interests include information security and privacy, distributed network, and software vulnerability detection.

Yu Bao is currently an associated professor at Security Department of Computer Science and Information Technology Institute, China University of Mining and Technology. He received the PhD degree from Tongji University in 2011, the MS and BS degrees from China University of Mining and Technology in 2003 and 2000, respectively. His research interests include intrusion detection techniques and trusted computing in IoT and security and privacy in wireless networks.

Cheng Ji is currently a staff engineer at the School of Management, China University of Mining and Technology. He received the PhD, MS, and BS degrees from China University of Mining and Technology in 2014, 2008, and 2002, respectively. His research interests include mobile computation, heterogeneous networks, and safety management.

Xin Li is currently an associate professor of computer science at China University of Mining and Technology. He received the PhD degree from China Academy of Technology in 2012, the MS degree from China University of Mining and Technology in 2004, and BS degree from Tianjin University of Science and Technology in 2001. His research interests include rescue robotics, mobile robot navigation, nonlinear filtering, computer vision, and machine learning. He is also interested in robotic perception, 3D reconstruction, sensor fusion, and SLAM.