TSINGHUA SCIENCE AND TECHNOLOGY
ISSNll1007-0214 01/10 pp497–519
DOI: 10.26599/TST.2018.9010116
Volume 24, Number 5, October 2019

Specification and Verification of a Topology-Aware Access Control Model for Cyber-Physical Space

Yan Cao, Zhiqiu Huang*, Shuanglong Kan, Dajuan Fan, and Yang Yang

Abstract: The cyber-physical space is a spatial environment that integrates the cyber and physical worlds to provide an intelligent environment for users to conduct their day-to-day activities. Mobile users and mobile objects are ubiquitous in this space, thereby exerting tremendous pressure on its security model. This model must ensure that both cyber and physical objects are always handled securely in this dynamic environment. In this paper, we propose a systematic solution to be able to specify security policies of the cyber-physical space and ensure that security requirements hold in these policies. We first formulate a topology configuration model to capture the topology characteristics of the cyber and physical worlds. Then, based on this model, a Topology-Aware Cyber-Physical Access Control model (TA-CPAC) is proposed, which can ensure the security of the cyber and physical worlds at the same time by adjusting permission assignment dynamically. Then, the topology configuration and TA-CPAC models are formalized by bigraphs and Bigraph Reactive System (BRS), respectively, allowing us to use model checking to rationalize the consequences of the evolution of topological configurations on the satisfaction of security requirements. Finally, a case study on a building automation access control system is conducted to evaluate the effectiveness of the proposed approach.

Key words: cyber-physical space; topology configuration; access control; model checking; bigraphs

1 Introduction

Computing and communication capabilities are increasingly embedded into physical spaces, thereby blurring the boundary between computational and physical worlds. This phenomenon is reflected by the notion of a cyber-physical system, where computational elements heavily interact with physical entities to monitor the behaviors of physical processes and the actuating actions to change their behaviors in accordance with the corresponding physical environment. A Cyber-Physical Space (CPP)[1] is a special case of the cyber-physical system; this space brings computation into the physical world to provide an intelligent spatial environment for roaming users, i.e., the smart building. In the CPP, users and resources are highly mobile in the physical and cyber worlds, which may result in ineffectiveness of the existing security configuration. An urgent problem is how to design adaptive and highly dependable technologies to ensure that cyber and physical resources can always be handled securely in different topology environments. This study focuses on the access control challenge which is critical among the numerous security challenges facing the CPP, including authentication,

• Yan Cao, Zhiqiu Huang, Shuanglong Kan, and Yang Yang are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. E-mail: caoyan926@nuaa.edu.cn; zqhuang@nuaa.edu.cn; kangshuanglong@nuaa.edu.cn; yychopper@163.com.
• Dajuan Fan is with the School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China. E-mail: fanbingjie523@126.com.
* To whom correspondence should be addressed.
Manuscript received: 2018-04-22; revised: 2018-08-15; accepted: 2018-09-01

reliability, denial-of-service attacks, etc.

Access control is an important information security technology that manages requests from users to access system resources. The security architecture of the open system interconnection reference model (ISO7492-2), which is proposed by the International Organization for Standardization, states that the access control service is an important component of security services and plays an irreplaceable role in the security architecture. The literature is rich in accounts of access control models[2–11]. However, to the best of our knowledge, no prior studies have considered the effect of the dynamic topology environment on the system access control model, including cyber and physical topologies.

First, the CPP is not a static paradigm[12]. The dynamic actions of entities (e.g., movements of subjects and objects, dynamic network communication) occur continuously. These actions change the topology of the cyber and physical spaces and may affect security concerns of the system. For example, the bank teller alone in the president's office may be a security threat to confidential files. Only when the president is in the office can the teller enter. The president can access the internal network only when he/she is not communicating with the external network. Therefore, the access control mechanism for the CPP should be able to perform security policies adaptively to react to topology changes in the physical and cyber spaces. However, the existing access control models for cyber-physical systems are only concerned with topology information in the physical space, such as location of subjects[3, 6] and proximity relationships among subjects[4].

Second, the CPP integrates computing and communication capabilities into physical processes. This integration enhances the intelligence of physical systems, but it also complicates the security problems because it involves the cyber security, physical security, and the interactions security at the same time[13]. For example, the president cannot exit the bank when he/she has confidential files on his/her computer (cyber-enabled physical access control). When a patient is in the office, the doctor cannot read the records of other patients (physical-enabled cyber access control). Therefore, a unified access control model that integrates physical access control and cyber access control is required in order to efficiently manage and implement security operations in this complex environment.

Third, the correctness of implementation of an access control model is based on the premise that the access control model is conflict free and conforms to the predefined security requirements[14]. At present, many organizations fall short of implementing the correct policies. For example, in large organizations, 50% – 90% of employees have over-entitled access, which presents opportunities for insiders to cause harm[15]. To design robust security policies for mission-critical systems, many researchers have addressed the issue of access control policy verification using various formal methods, e.g., NuSMV[16], UML/OCL[6], Alloy[11], Colored Petri Nets (CPN)[3], and ambient calculus[17]. All of these methods, except ambient calculus, cannot describe the topology of the physical and cyber spaces intuitively, thereby complicating the modeling process. The ambient calculus represents the topology of the physical space to help identify and prevent potential future violations of security requirements, but the ability to represent the changes in cyber topology (e.g. networks that have a dynamic topology) is limited. Therefore, an appropriate method is necessary to check the access control policies of the CPP and detect security breaches in them.

In this study, we develop an effective access control model and a rigorous security breach detection method for the CPP. First, a topology configuration model is proposed, which accounts for the key structure and relationships of the operational environment. Considering this model, we can understand how topological changes affect security concerns. Second, a Topology-Aware Cyber-Physical Access Control (TA-CPAC) model is presented. This model takes the Role-Based Access Control (RBAC, ANSI INCITS 359-2004) model as the main framework, and the topology attributes are added to the RBAC model to support the dynamic permission assignment. These attributes include location of subjects and objects, proximity relationships among entities, activation states between users and roles, etc. Third, to ensure highly dependable access control policies, we select bigraphs and Bigraph Reactive System (BRS) as the semantic domain to perform the verification of security properties. Bigraphs specify the topology configuration model of the CPP. The BRS specifies actions defined in access control policies. If an action is allowed by the current topology configuration, it is executed and the configuration is updated accordingly. The security properties are also expressed by bigraphs, and their validity is checked by

the model checking approach. A reduction algorithm is proposed to reduce the search spaces in the process of model checking. Finally, we evaluate our approach through a case study to demonstrate its effectiveness and feasibility.

The main contributions of this paper are summarized as follows.

(1) A topology configuration model is proposed to describe the topology of the operational environment.

(2) A TA-CPAC model for the CPP is proposed. This model can ensure that cyber and physical resources are handled securely no matter how the topology of the operational environment changes.

(3) Bigraphs and BRS are used to represent the topology of a cyber-physical space and rationalize the consequences of topological changes on the satisfaction of security requirements.

(4) A reduction algorithm is proposed to simplify the modeling process.

The rest of the paper is organized as follows. Section 2 summarizes the related work. In Section 3, the topology configuration and TA-CPAC models are defined. In Section 4, transformation rules from the topology configuration and TA-CPAC models to formal methods are presented. In Section 5, our method is validated using a building automation access control system. Finally, in Section 6, we present a summary of our work and highlight directions for future research.

2 Related Work

The cyber-physical space is becoming increasingly pervasive across the critical infrastructure. The access control model for this space has drawn close attention in recent years. Chen et al.[2] proposed a reputation-based access control model called R2BAC, which assigns roles to users based on the reputation evaluation of their past behaviors. Toahchoodee and Ray[3] proposed a spatio-temporal role-based access control model where authorization decision depends on the role of the user, the locations of the subjects and objects, and the time of access. Kirkpatrick et al.[4,5] presented a prox-RBAC model where the access control decision is not only based on the requesting user's location but also considers the location of other users in the system. Fadhel et al.[6] proposed a comprehensive framework GemRBAC model that expresses the various types of RBAC policies, including location-based policies, time-based policies, and so on. Huang et al.[7] proposed a two-layer access control model by integrating attributes into RBAC. The aboveground level is a standard RBAC model extended with environment. The underground level is used to represent security knowledge in terms of attribute-based policies, which are easy to build and can easily adapt to changes. Jin et al.[8] proposed a novel Role-centric Attribute-Based Access Control (RABAC) model that extends the RBAC model with permission filtering policies. These filtering policies constrain the available set of permissions based on user and object attributes. All these models use the physical context information (e.g., location of subjects, proximity relationships among subjects) to make decisions on access requests that occur in cyber-space. In other words, they only focus on cyber security. Skandhakumar et al.[9] proposed a method that utilizes Building Information Models (BIMs) to reduce the incidents of error in the physical space. Turkmen et al.[10] concerned with computing relaxations of physical access policies to eliminate conflicting rules. Geepalla et al.[11] proposed an STRBAC-PS model that considers the physical topology aspects in access control systems. A location graph is proposed in this model to formalize the physical location. All these models focus on physical security. For the convergence of physical and cyber access control, Unal and Caglayan[18] proposed an FPM-RBAC model that concerns the location, proximity, and mobility aspects of entities in mobile networks. This model considers the interplay between the physical and cyber spaces but is not concerned with the connectivity state between subjects and objects, a key characteristic of cyber-space.

To ensure that an access control system is safe, a reliable means is needed to verify that the specified access control policies conform to the security requirements and policy author's intentions prior to their implementation. Jha et al.[19] performed a comparison between the use of the model checking and the first-order logic programming for the security analysis of access control policies and conclude that the model checking is a promising method. Hu et al.[16] verified the integrity, coverage, and confinement properties of access control policies using Access Control Policy Testing System (ACPTS) tool, which is based on NuSMV and Automated Combinatorial Testing Suite (ACTS) tools. Gouglidis et al.[20] proposed the verification of secure inter operation properties for RBAC systems using NuSMV tool. These two verification works do not consider the physical elements. The GemRBAC model proposed

in Ref. [6] is formalized by UML/OCL to eliminate conflicts. UML/OCL is a semi-formal method that cannot verify behavioral properties. Toahchoodee et al.[21] used Alloy to verify spatio-temporal RBAC policies, but Alloy faces difficulty in deciding whether a policy holds or not when the Alloy analyzer cannot find a counterexample of a property in question within a certain scope. Some researchers use CPN and Time Automata (TA) approaches to verify spatio-temporal RBAC policies[3, 22]. The location information in the CPN is represented by tokens of a product color of string variables. In the TA, an entity at a particular location zone is represented as a time-automata. These two methods easily cause a drastic increase in the model state-space. Unal and Caglayan[17] used the ambient calculus to specify the current state of a mobile network and actions within security policies. The ambient calculus encodes computation only as process-algebraic structural changes in a hierarchy and hinders reasoning about communication and links in cyber-space. Pasquale et al.[23] proposed a visualization tool for a smart space that allows security analysts to edit the space topology and verify whether access control policies meet security requirements, but this method is based on the breadth-first graph search algorithm, which does not concern security properties, such as the separation-of-duty. All the methods introduced are not supportive of mechanisms to reason about topological characteristics of the CPP affected by both structures of the spaces as well as by links.

Bigraphs and BRS are an emerging formalism for describing static spatial and communication relationships alongside a set of reaction rules that defines dynamic behaviors. They have been considered extensively in studies on modeling cyber and physical spaces. Walton and Worboys[24] used bigraphs to support the description of scenes and narratives with incomplete information, and provided a set reactions rules dictating legal system transformations to support goal-directed navigation. Tsigkanos et al.[25] used them to support property verification of the smart building adopting software engineering principles. Benford et al.[26] used expanded BRS to explain observed inconsistencies in user trials of Savannah game, and reveal an incompleteness in design. By contrast, we use bigraphs and BRS to analyze the correctness of topology-aware access control policies in this study.

We compare existing studies on policy specification and policy verification, and highlight our work in Table 1.

3 Topology-Aware Access Control for Cyber-Physical Space

In this section, we first propose a topology configuration model to express the topology configuration of the operational environment. Then, the TA-CPAC model is proposed. The separation of duty constraints is supported in this model. Finally, authorization term is defined to specify security policies.

3.1 Topology configuration description

3.1.1 Topology of cyber-physical space

Before explaining the topology of the CPP in detail, we define some basic terms. Objects (also known as assets) in the cyber-physical space are classified as physical objects, hybrid objects, and cyber objects. Physical objects are hardware devices equipped with sensors or actuators. Cyber objects are digital entities located in hybrid objects. Hybrid objects can be conceived as having physical and cyber characteristics, which delimit digital areas for cyber objects. The concept of entities

Table 1 Comparison of related works.

Model	Reference	Policy specification					Policy verification	
		CS	PS	MS	MO	SO	FD	VT
R2BAC	[2]	√	×	×	×	×	×	×
Spatio-temporal model	[3, 21]	√	×	√	√	×	Petri net or Alloy	CPN or Alloy analyzer
Prox-RBAC	[4]	√	×	√	×	×	×	×
GemRBAC	[6]	√	×	√	×	×	UML/OCL	USE
Two-layered access control model	[7]	√	×	√	×	×	×	×
RABAC	[8]	√	×	√	×	×	×	×
Access control model with BIM	[9]	×	√	√	×	×	×	×
Physical access control model	[10]	×	√	√	×	×	×	×
STRBAC-PS	[11]	×	√	√	×	×	Alloy	Alloy analyzer
FPM-RBAC	[17, 18]	√	√	√	√	×	Ambient calculus	Java language and NuSMV
Our work	This paper	√	√	√	√	√	Bigraphs and BRS	BigMC

Notes: √: support; ×: no support; CS: cyber security; PS: physical security; MS: mobile subjects; MO: mobile objects; SO: communication between subjects and objects; FD: formal description; VT: verification tool.

includes objects and users that are in the cyber-physical space.

The topology refers to the structure in terms of the key elements and their relationships that determine the shape of the environment[27]. In a physical sense, a topology denotes characteristics of the physical space, such as deployment of the building, and location of users. In cyber-space, a topology denotes the structural characteristics of information, such as location of cyber objects, and users accessing behaviors to objects. For example, Fig. 1 shows the topology configuration of a smart bank. The bank branch has a main area in which customer businesses are handled and from which private areas of the bank can be accessed, including a server room, a client manager's office, a teller's office, a president's office, and an accountant's office. A safe room is in the president's office. The safe and box are the physical objects. The hybrid objects include a cloudlet, a server, and mobile phones. The cyber objects include file1, file2, and file3. The locations of entities are shown in this figure intuitively. The communication relationship is described by dotted lines. Alice logs in the server in the president's office.

The entities change their states continuously, which may render the existing configuration ineffective. For example, Bob can copy data from Alice, Bob can enter the president's office, and Alice can take file2 in the president's office. If the security requirement is that Bob does not have the right to get file2, then the above actions can bring the topology configuration into a violate state. Knowing where valuable assets are placed and their relationships with other entities is helpful to identify security controls that can protect those assets. Thus, the topology configuration of the operational environment is an important factor in achieving adaptive access control for the CPP.

3.1.2 Topology configuration model

The formal definition of the topology configuration should include all the elements and relationships introduced above.

Definition 1 Topology configuration model. The topology configuration model is a 12-tuple:

$CE = \langle User, Role, Object, Lloc, Loco, UserRoleassign, UserRoleactivate, RoleObject, ObjectLocation, Locrelation, RoleLocation, UserLocation\rangle$.

(1) *User, Role,* and *Object* stand for the set of users, roles, and objects that are in the cyber-physical space, respectively.

(2) *Lloc* and *Loco* stand for the set of space locations and object locations, respectively.

(3) $UserRoleassign \subseteq User \times Role$ indicates the set of assignment relationships between users and roles in the system.

(4) $UserRoleactivate \subseteq User \times Role$ indicates the set of activation relationships between users and roles in the current environment.

(5) $UserLocation \subseteq User \times Lloc$ indicates the set of user-location relationships in the current environment.

(6) $RoleLocation \subseteq Role \times Lloc$ indicates the set of role-location relationships in the current environment.

(7) $ObjectLocation \subseteq Object \times Loco$ indicates the set of object-location relationships in the current environment.

(8) $RoleObject \subseteq Role \times Object$ indicates the set of access behaviors between subjects and objects in the current environment.

(9) $Locrelation \subseteq Lloc \times Lloc$ indicates the reachability relationships of the physical spaces.

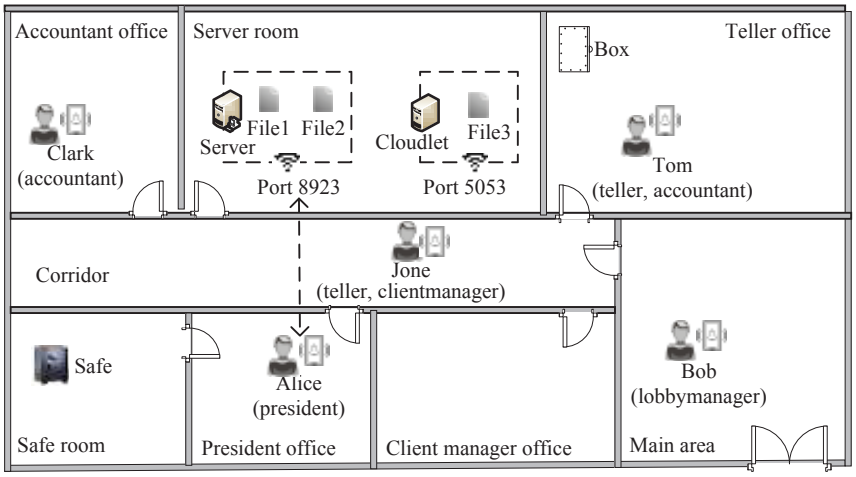


Fig. 1 Deployment of bank branch.

$(lr_i, lr_j) \in Locrelation$ specifies that from the entrance area of the building, we can go through location lr_i to location lr_j.

The set $Lloc = \{lr_1, lr_2, \ldots, lr_n | n \in \mathbf{N}\}$ represents physical locations of the building, such as the main area and the teller's office. The set of *Locrelation* is taken from the deployment of the building. The user-role assignment is specified by the administrator of security policies. Thus, *Locrelation* and *UserRoleassign* sets are static and the same in all topology configurations. Other sets vary in different environments based on the actions that have already been executed. We define these dynamic elements in the topology constraint set C, $C = \{UserRoleactivate, UserLocation, RoleLocation, ObjectLocation, RoleObject\}$. This set supports the definition of the access control model, which is elaborated in Section 3.2.

In addition, the descriptions of the location of objects vary for different types of objects. For physical or hybrid objects, the location is a member of the location set *Lloc*. As cyber objects are located in hybrid objects, the location of cyber objects consists of two layers: first is the hybrid object and second is the location of this hybrid object. For example, the first layer location of file1 in Fig. 1 is the server, and the second layer location is the server room. The topology configuration model of Fig. 1 is defined in Appendix Table A1.

3.2 Specification of the topology-aware cyber-physical access control policy

3.2.1 TA-CPAC model

The topology configuration model only describes the static characteristics of the operational environment. The dynamic behaviors are defined in the TA-CPAC model, which is depicted in Fig. 2.

As users and objects are in the roaming state, their location is an important parameter in security policies. The data set consists of the following six basic elements: users, roles, operations, objects, Lloc, and Loco. The latter four form a new element of permissions. In addition, the space locations are included in the set objects because the behaviors of entering and exiting a location in the physical space are the key management point. The relation set includes the Role enabling (RL), Object enabling (OL), Permission enabling (PL), User-role Assignment (UA), User-role Enabling (UE), and role-Permission Assignment (PA). For a location, the visitor and executable permissions are both constrained. These relationships are described by the RL relation and PL relation, respectively. Users are assigned a set of roles, and can enable different roles in different locations, which are described by UA and UE relations, respectively. Roles are associated with different permissions in different locations, described by the PA relation. The concept of user sessions is a mapping of one user to some subset of enabled roles according to the user's location, allowing the user selective activation and deactivation of these enabled roles. Aside from the basic concepts, functions are defined to describe the relationships among the basic concepts.

Definition 2 TA-CPAC model. The TA-CPAC model is defined as

(1) $Users, Roles$, $Operations$, $Loco$, and $Lloc$ stand for the set of users, roles, operations, object locations, and space locations, respectively.

(2) $UA \subseteq Users \times Roles$ is the user-role assignment, a many-to-many mapping user-to-role assignment relation.

- $assigned\_user(r : Roles) \rightarrow 2^{Users}$ is the mapping

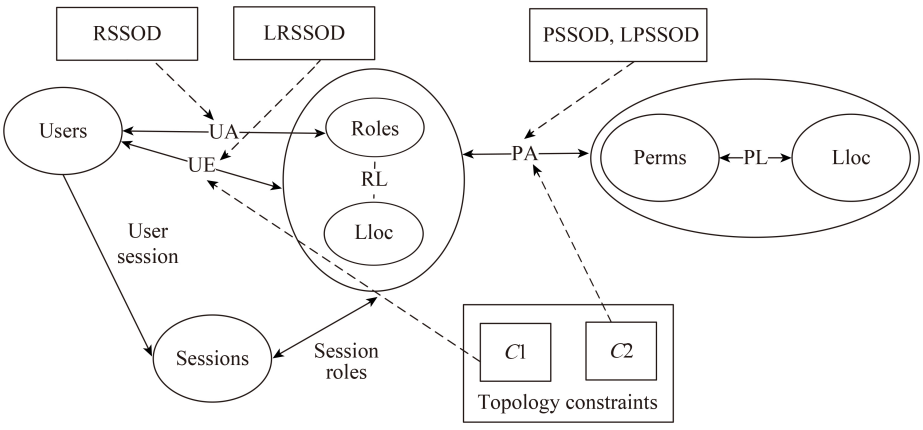


Fig. 2 TA-CPAC model.

of a role r onto a set of users. Formally, $assigned\_user(r) = \{u \in Users | (u, r) \in UA\}$.

(3) $RL \subseteq Roles \times Lloc$ represents role enabling, a many-to-many mapping role-to-location relation.

- $LlocRole(lr : Lloc) \rightarrow 2^{Roles}$ is the mapping of a location lr onto a set of roles. These roles are called enabled roles for the location lr. Formally, $LlocRole(lr) = \{r \in Roles | (r, lr) \in RL\}$.

- $RoleLloc(r : Roles) \rightarrow 2^{Lloc}$ is the mapping of a role r onto a set of locations, where the role r can be enabled. Formally, $RoleLloc(r) = \{lr \in Lloc | (r, lr) \in RL\}$.

(4) $UE \subseteq Users \times RL$ is the user-role enabling relation.

- $enabled\_user(r : Roles, lr : Lloc) \rightarrow 2^{Users}$ is the mapping of an enabled role r in the location lr onto a set of users. Formally, $enabled\_user(r, lr) = \{u \in Users | (u, r) \in UA \wedge (r, lr) \in RL\}$.

- $enabled\_role(u : Users, lr : Lloc) \rightarrow 2^{Roles}$ is the mapping of a user u in the location lr onto a set of roles. Formally, $enabled\_role(u, lr) = \{r \in Roles | (u, r) \in UA \wedge (r, lr) \in RL\}$.

(5) $Objects = \{Physicalobject, Hybridobject, Cyberobject, Lloc\}$ represents the set of objects to be protected consists of physical objects, hybrid objects, cyber objects, and space locations.

(6) $OL \subseteq Objects \times Loco$ represents object enabling, a many-to-many mapping object-to-location relation.

(7) $Perms = 2^{(Operations \times OL)}$ is the set of permissions.

(8) $PL \subseteq Perms \times Lloc$ is permission enabling, a many-to-many mapping permission-to-location relation.

- $LlocPerm(lr : Lloc) \rightarrow 2^{Perms}$ is the mapping of a location lr onto a set of permissions. These permissions are called enabled permissions for the location lr. Formally, $LlocPerm(lr) = \{p \in Perms | (p, lr) \in PL\}$.

- $PermLloc(p : Perms) \rightarrow 2^{Lloc}$ is the mapping of a permission p onto a set of locations, where the permission p can be enabled. Formally, $PermLloc(p) = \{lr \in Lloc | (p, lr) \in PL\}$.

(9) $PA \subseteq Roles \times PL$ is the permission-role assignment, a many-to-many mapping permission-to-role assignment relation.

- $can\_AssignPerm(r, p, lr) \Rightarrow r \in Roles \wedge p \in Perms \wedge lr \in (RoleLloc(r) \cap PermLloc(p))$. When the role r can execute the permission p in the location lr, this predicate returns *TRUE*. In the location lr, the role r and the permission p are both enabled.

- $enabled\_permission(r : Roles, lr : Lloc) \rightarrow 2^{Perms}$ represents the mapping of a role r onto a set of permissions which are assigned to this role in the location lr. These permissions are called enabled permissions for the role r in the location lr. Formally, $enabled\_permission(r : Roles, lr : Lloc) = \{p \in Perms | can\_AssignPerm(r, p, lr)\}$.

- $assigned\_permission(r : Roles) \rightarrow 2^{Perms \times Lloc}$ is the mapping of a role r onto a set of permissions which are assigned to this role. Formally, $assigned\_permission(r : Roles) = \{(p, lr) \in PL | can\_AssignPerm(r, p, lr)\}$.

(10) *Sessions* indicates the set of sessions.

- $session\_user(s : Sessions) \rightarrow Users$ is the mapping of a session s onto a corresponding user.

- $session\_role(s{:}Sessions, lr{:}Lloc) \rightarrow 2^{Roles}$ represents the mapping of the session s onto a set of roles in the location lr. Formally, $session\_role(s, lr) \subseteq enabled\_role(session\_user(s), lr)$.

(11) $C1 = \{UserRoleactivate, UserLocation, ObjectLocation\}$ is the topology constraint set which restricts the UE relation. $C2 = \{ObjectLocation, RoleLocation, RoleObject\}$ is the topology constraint set which restricts the PA relation.

3.2.2 Separation of duty constraints on TA-CPAC model

Separation of Duty (SoD) constraints are used to enforce conflict of interest policies that organizations may employ to minimize the likelihood of collusion among individuals. In other words, SoD serves as a requirement for critical operations, which are divided among two or more subjects, so that no single subject can compromise security. Two types of SoD are involved in the TA-CPAC model: one is to restrict the user-role relation, and the other is for the role-permission relation.

Definition 3 Role Static Separation Of Duty (RSSOD) and Location-Related Static Separation Of Duty (LRSSOD). RSSOD and LRSSOD are specified as follows:

(1) $RSSOD \subseteq 2^{Roles} \times N$ is a collection of conflicting roles, $(rs, n) \in RSSOD$, where rs is a subset of roles and n is a natural number, $n \geqslant 2$, with the property that no user is assigned to n or more roles from the set rs. Formally, $\forall (rs, n) \in RSSOD, \forall t \subseteq rs : |t| \geqslant n \Rightarrow \bigcap_{r \in t} assigned\_user(r) = \varnothing$.

(2) $LRSSOD \subseteq 2^{Roles \times Lloc} \times N$ is a collection of conflicting roles associated with their enabled locations, $(rl, n) \in LRSSOD$, where rl is a subset of RL and n is a

natural number $n \geqslant 2$, and no existing relationship that n or more roles from the set rl are enabled for a user. Formally, $\forall (rl, n) \in LRSSOD, \forall t \subseteq rl : |t| \geqslant n \Rightarrow \bigcap_{(r,lr)\in t} enabled\_user(r, lr) = \varnothing$.

Definition 4 Permission Static Separation Of Duty (PSSOD) and Location-related Permission Static Separation Of Duty (LPSSOD). PSSOD and LPSSOD are specified as follows:

(1) $PSSOD \subseteq 2^{Perms} \times N$ is a collection of conflicting permissions. $(ps, n) \in PSSOD$, where ps is a subset of permissions and n is a natural number, $n \geqslant 2$, with the property that no role is assigned to n or more permissions from the set ps. Formally, $\forall r \in Roles, (ps, n) \in PSSOD, \forall t \subseteq ps : |t| \geqslant n \Rightarrow t \not\subset \bigcup_{lr\in Lloc} enabled\_permission(r, lr)$.

(2) $LPSSOD \subseteq 2^{Perms\times Lloc} \times N$ is a collection of conflicting permissions associated with locations. $(pl, n) \in LPSSOD$, where pl is a subset of PL and n is a natural number, $n \geqslant 2$, with the property that no role is assigned to n or more permissions from the set pl. Formally, $\forall r \in Roles, (pl, n) \in LPSSOD, \forall t \subseteq pl : |t| \geqslant n \Rightarrow t \not\subset assigned\_permission(r)$.

RSSOD is a constraint on the UA relation. It determines the set of conflicting roles that should not be assigned to the same user. PSSOD is a constraint on the PA relation to determine conflicting permissions that cannot assign to the same role. For example, $(r1, r2, 2) \in RSSOD$ states that a user assigned to role $r1$ cannot be assigned to role $r2$ or vice versa. $(p1, p2, 2) \in PSSOD$ states that a role has permission $p1$, and then this role cannot be given permission $p2$ or vice versa. LRSSOD and LPSSOD are for supporting the location-restricted SoD. For example, $((r1, lr1), (r2, lr1), 2) \in LRSSOD$ states that the roles $r1$ and $r2$ cannot be in enabled states at the same time in the location $lr1$. $((p1, lr1), (p2, lr2), 2) \in LPSSOD$ states that a role has permission $p1$ at location $lr1$, and then this role cannot be assigned to permission $p2$ at location $lr2$ or vice versa. RSSOD and PSSOD constraints are not constrained by the location. In other words, these constraints are valid for all locations.

3.2.3 Authorization term

To summarize the aforementioned policy elements, we define the authorization term, which is the basic formal construct used to specify access control policies.

Definition 5 Authorization term. An Authorization Term (AT) is a tuple $(u, r, op, o, lo, lr, c, c')$, where $(u, r, lr) \in UE$, $(r, op, o, lo, lr) \in PA$.

(1) $u \in Users$ represents a user who puts forward the access request.

(2) $r \in Roles$ indicates the role of the user.

(3) $op \in Operations$ is the operation executed by the user. For this study, we assume a fixed set of operations $Operations = \{enter, exit, open, close, login, logout, copy, delete\}$. More operations can be added if required.

(4) $o \in Objects$ is the object to be accessed.

(5) $lo \in Loco$ is an enabled location of the object.

(6) $lr \in Lloc$ is the physical location in which the user puts forward the access request.

(7) c and c' are the topology constraint formulas on UE and PA relations, respectively. They determine the applicability of the authorization term.

The topology constraint formula c is obtained based on Boolean operations on the set $C1$.

Definition 6 Topology constraint formula. The topology constraint formula c is defined as follows:

(1) ϕ is a topology constraint formula.

(2) If p is an element of the set $C1$, it is a topology constraint formula.

(3) If p and q are the topology constraint formula, so are $p \wedge q$, $p \vee q$, and $\neg p$.

The topology constraint formula c' is defined on the set $C2$. We omitted this definition for simplicity. These formulas can support four types of topology constraints, including activation, location, proximity, and communication constraints. (1) Activation constraints specify the activated roles of the user. (2) Location constraints specify the location of users and objects. (3) Proximity constraints specify the location relationships among entities, including user proximity constraints, object proximity constraints, and user and object proximity constraints. (4) Communication constraints specify the access behaviors between users and objects. If the formula c is ϕ, then the enabling of the role is not affected by topology constraints. If the formula c' is ϕ, then the execution of the permission is not affected by topology constraints.

Example 1 We present some examples to explain the authorization terms.

(1) John can activate the role "student" in the classroom after Alice has activated the role "teacher" in this room.

(2) When the patient records are in the doctor's office, the cleaner cannot enter this office (from the corridor).

(3) The manager in the main area cannot log in the server and cloudlet simultaneously.

(4) When the president is in the president's office, the

teller can log in the server in this office.

The type of the first item is the UE relation with activation constraints. The second item is the PA relation with location constraints. The third item is the PA relation with communication constraints. The last item is the PA relation with proximity constraints. The formal representations are as follows:

(1) (*John, student, classroom, UserRoleactivate* (*Alice, teacher*)∧ *UserLocation* (*Alice, classroom*)).

(2) (*cleaner, enter, doctoroffice, ϕ, corridor,* ¬*Object-Location* (*records, doctor office*)).

(3) (*manager, login, server, serverroom, mainarea,* ¬ (*RoleObject* (*manager, cloudlet*) ∧ *ObjectLocation* (*cloudlet, serverroom*) ∧ *RoleLocation* (*manager, mainarea*))).

(4) (*teller, login, server, serverroom, presidentoffice, RoleLocation* (*president, presidentoffice*)).

4 Security Analysis of TA-CPAC Model Using Model Checking

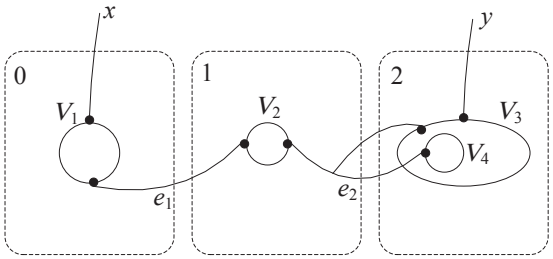
In this section, we provide a formal framework for policy administrators to verify security policies prior to their implementation. Initially, the formal methods of bigraphs and BRS are introduced. Then, the transformation rules from the topology configuration model to bigraphs and the TA-CPAC model to BRS are proposed. Next, security properties are extracted from the security requirements and expressed by bigraphs. The validity of the security property is checked through model checking. A reduction algorithm is proposed in the last subsection to reduce the search space in the model checking process.

4.1 Bigraphs and BRS

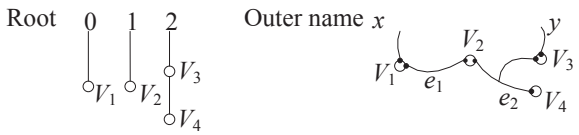
A modeling formalism for the CPP should allow the representation of the topology of the operational environment. It should also enable reasoning about the effects of topological changes arising from execution actions defined in security policies. We use bigraphs and BRS that have a number of features which are convenient for formal specifications of our models. First, bigraphs consider both the structure and linking, which help in modeling the topology configuration. Second, BRS defines a set of reaction rules that allows reasoning about possible future states that are reachable from the current state. Thus, the actions defined in access control policies can be modeled through reaction rules, which are helpful for analyzing whether the execution of actions can lead to a violated topology configuration. Third, bigraphs and BRS not only have a complete axiomatic system but also provide graphical representations which are beneficial to the user's understanding.

Bigraphs consist of a place graph, a forest defined over a set of nodes which is intended to represent entities and their locality in terms of a containment structure, and a link graph, a hypergraph composed over the same set of nodes representing arbitrary linking among those entities. Connections of an edge with its nodes are called ports. The type of nodes, called controls, is defined by a so-called signature. Figure 3 shows a bigraph structure. The upper part is a bigraph F, and the lower part is its corresponding place graph and link graph. V_1, V_2, V_3, and V_4 are nodes shared between the place graph and the link graph. The nodes are nested to express the location containment. The links include closed links, such as e_1, e_2, and open links, such as the edge between V_1 and x, V_3 and y. x and y are called outer names, which are expressed by using links reaching to the top. Links reaching to downward are called inner names. In addition, bigraphs also have a special class of nodes called sites, which are used to abstract the things we are not concerned with, and are identified by the gray dotted frame.

Bigraphs represent the static aspects of systems. The dynamic aspects are expressed by BRS, which includes a set of reaction rules. These rules present possible ways in which a system might be reconfigured. A reaction rule has the form $R \rightarrow R'$, where R is known as the redex and R' as the reactum. The redex and reactum are both bigraphs. The intuitive definition of the reaction rule is fairly straightforward. If we can find



(a) Bigraph $F: \varepsilon \rightarrow \langle 3, \{xy\} \rangle$



(b) Place graph $F^P: 0 \rightarrow 3$ (c) Link graph $F^L: \varnothing \rightarrow \{xy\}$

Fig. 3 Bigraph *F* structure diagram.

an instance of the redex somewhere in a bigraph S, then we may replace the redex with the reactum to obtain a new bigraph S' in a manner similar to graph rewriting.

The preceding graphical representation is useful to model but unwieldy for reasoning. Fortunately, Milner[28] proposed the algebra description for bigraphs and BRS, which is called term languages. A part of the language is summarized in Table 2. U and V are controls of nodes. Nodes can be structured hierarchically through the nesting relationship. Two nodes or regions at the same hierarchical level are expressed by the juxtaposition relationships. The site node is expressed by the letter \$. In the last formula, the node identified by the control K and port names in list w also contains U. Ports that appear in a formula with the same name are connected. The interested reader can refer to the work by Milner[28] for complete definitions and proofs of the theory.

4.2 Mapping from topology configuration model to bigraphs

By using the place graph and link graph defined in bigraphs, we can model the topology configuration in a natural manner. We first introduce the formal definition of bigraphs for the CPP and then discuss the transformation rules from the topology configuration model to the bigraphs.

Definition 7 Bigraphs for CPP. A concrete bigraph for CPP $B = (V_B, E_B, ctrl_B, prnt_B, link_B) : \langle m, Y\rangle \to \langle n, Y\rangle$ consists of a concrete place graph $F^P = (V_B, ctrl_B, prnt_B) : m \to n$ and a concrete link graph $F^L = (V_B, E_B, ctrl_B, link_B) : X \to Y$.

(1) $F^P = (V_B, ctrl_B, prnt_B) : m \to n$, where an inner face m and an outer face n are the numbers of sites and regions, respectively. F^P has a finite set of nodes V_B, a control map $ctrl_B : V_B \to K$, and a parent map $prnt_B : m \uplus V_B \mapsto V_B \uplus n$. "$\mapsto$" expresses that the tail node is located in the direction node. K is the set of signatures.

(2) $F^L = (V_B, E_B, ctrl_B, link_B) : X \to Y$, where X and Y are the set of inner names and outer names of the link graph. F^L has a finite set of nodes V_B, a finite set of edges E_B, a control map $ctrl_B : V_B \to K$, and a link map $link_B : X \uplus P_B \leftrightarrows E_B \uplus Y$. P_B is a set of ports of the node.

Table 2 Term languages for bigraphs.

Term language	Definition
$U.V$	Nesting (U contains V)
$U \| V$	Juxtaposition of regions
$U \mid V$	Juxtaposition of nodes
$\$i$	Site number i
$K_w.(U)$	Node associated with control K having ports with names in w. K contains U.

In the TA-CPAC model, users obtain permissions by being members of roles. The correctness of UE and PA relations is the key to obtain a correct set of authorization terms. For the security analysis of the UE relation, there is no need to consider permissions. Furthermore, users need not be considered for the security analysis of the PA relation. Therefore, different models are established for different relations. Transformation rule 1a converts the topology configuration model to bigraphs for analysis of the UE relation. The users, roles, objects, and space locations are expressed by nodes. The locations of subjects and objects, UA relation, and reachability relationships in the physical spaces are all expressed by the node nesting. The role, user, and location nodes are nested sequentially. The link graph represents the user-role activation relation by linking port names. Transformation rule 1b converts the topology configuration model to bigraphs for analysis of the PA relation. The role node is directly nested in the location node. The role-accessing behaviors are expressed by link relationships. In these two transformation rules, the outer names are instantiated according to the specific application. The number of region is one because we only consider a single administrative domain in this study. The set K defines the signatures of controls. The number of ports for one control and their meaning depends on the system. For our model, the port number of the role node is one. In Transformation rule 1a, this port connects the user's port to express the user-role activation state. In Transformation rule 1b, this port links the object's port to express the role-accessing state.

Example 2 Based on the transformation rules, the graphical bigraphs representations of the bank topology configuration model are shown in Fig. 4. In general, bigraphs permit any kind of shape (sometimes colors) for entities. In this figure, the circle, triangle, hexagon, ellipse, rectangle, and rounded rectangle stand for roles, cyber objects, physical objects, hybrid objects, locations, and users, respectively. As the mobile phone is private, the subject nodes are used instead of the phone nodes for simplicity. Figure 4a is one configuration of the bank system for analyzing the UE relation. The corresponding term language is

Transformation rule 1a: Transformation rule from the topology configuration model to bigraphs for analyzing the UE relation

V_B : $User, Role, Object, Lloc \Rightarrow V_B$ /* Users, Roles, Objects, space locations map to nodes*/
$ctrl_B : V_B \rightarrow K$ /*control map*/
$prnt_B : (lr_i, lr_j) \in Locrelation \Rightarrow lr_j \mapsto lr_i$
/*reachability relationships in the physical space map to location containment*/
$(o, lo) \in ObjectLocation \Rightarrow o \mapsto lo$
/*object-location relationships map to location containment*/
$(u, r) \in UserRoleassign \wedge (u, lr) \in UserLocation \Rightarrow r \mapsto u \wedge u \mapsto lr$
/* subject-location relationships map to location containment. */
$link_B : (u, r) \in UserRoleactivate \Rightarrow P_B(u) \leftrightarrows e_i \wedge P_B(r) \leftrightarrows e_i$
/* user-role activation relationships map to the connectivity */
$P'_B \leftrightarrows Y$ /*idle ports link to the outer name*/
$E_B = \{e_1, e_2, \ldots, c_n\}$/*set of edges*/
$m = 0$ /*number of sites is zero*/
$n = 1$ /*number of region is one*/
$X = \varnothing$ /*inter name is empty*/
Y: set of outer names for user, role, and object nodes.
where : " $\Rightarrow$ " stands for transformation relationships.
$K = \{(User, 1), (Role, 1), (Lloc, 0), (Hybridobject, 1), (Physicalobject, 1), (Cyberobject, 0)\}$

Transformation rule 1b: Transformation rule from the topology configuration model to bigraphs for analyzing the PA relation

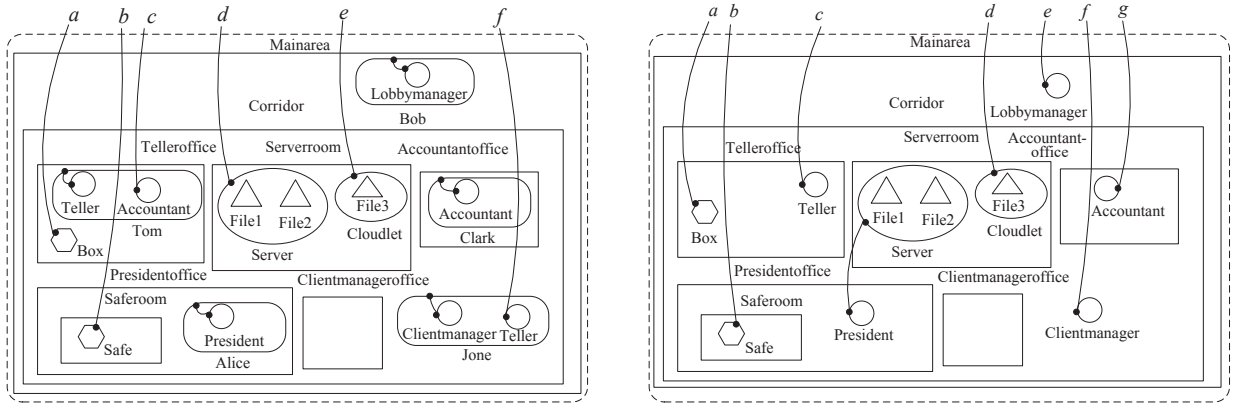
V_B : $Role, Object, Lloc \Rightarrow V_B$ /* Roles,Objects,space locations map to nodes*/
$ctrl_B : V_B \rightarrow K$ /*control map*/
$prnt_B : (lr_i, lr_j) \in Locrelation \Rightarrow lr_j \mapsto lr_i$
/*reachability relationships in physical space map to location containment*/
$(o, lo) \in ObjectLocation \Rightarrow o \mapsto lo$
/*object-location relationships map to location containment*/
$(r, lr) \in RoleLocation \Rightarrow r \mapsto lr$
/* subject-location relationships map to location containment. */
$link_B : (r, o) \in RoleObject \Rightarrow P_B(r) \leftrightarrows e_i \wedge P_B(o) \leftrightarrows e_i$
/* subject accessing behaviors map to the connectivity */
$P'_B \leftrightarrows Y$ /*idle ports link to the outer name*/
$E_B = \{e_1, e_2, \ldots, c_n\}$: /*set of edges*/
$m = 0$ /*number of sites is zero*/
$n = 1$ /*number of region is one*/
$X = \varnothing$ /*inter name is empty*/
Y: set of outer names for role and object nodes.
where : " $\Rightarrow$ " stands for transformation relationships.
$K = \{(Role, 1), (Lloc, 0), (Hybridobject, 1), (Physicalobject, 1), (Cyberobject, 0)\}$

$mainarea.\ (Bob_u.\ lobbymanager_u \mid corridor.\ (Jone_t.\ (clientmanager_t \mid teller_f) \mid accountantoffice.Clark_s.\ accountant_s \mid clientmanageroffice \mid presidentoffice.\ (saferroom.safe_b | Alice_w.\ president_w) | telleroffice.(box_a | Tom_v.(teller_v | accountant_c)) | serverroom.(server_d.(file1 | file2) | cloudlet_e.file3)))$. Figure 4b is one configuration for analyzing the PA relation. The term language is $mainarea.(lobbymanager_e \mid corridor.(clientmanager_f | accountantoffce.accountant_g | presidentoffice.(saferoom.safe_b | president_x) | clientmanageroffice | telleroffice.(box_a | teller_c) \mid serverroom.(server_x.(file1 \mid file2) \mid cloudlet_d.file3)))$. The letters a, b, c, d, e, f, and g are outer names of nodes. The nodes with the same subscript letters are connected by links.

4.3 Mapping from TA-CPAC model to BRS

Having defined how bigraphs provide static semantics of the CPP, we proceed to consider the actions defined in security policies, thereby giving rise to dynamic features. These actions are formalized by BRS.

Owing to the length limitation, the transformation templates in Transformation rule 2 only involve



(a) A topology configuration of the bank system for analyzing UE relation

(b) A topology configuration of the bank system for analyzing PA relation

Fig. 4 Bigraphs of bank topology configuration model.

Transformation rule 2: Transformation rule from authorization terms to reaction rules

UE:$(u, r, lr) \Rightarrow lr.(u_a.(r_b.\$0|\$1)|\$2) \rightarrow lr.(u_x.(r_x.\$0|\$1)|\$2)$

PA: $(r, enter, lr1, \phi, lr2) \Rightarrow lr2.(r_b.\$0|lr1.\$1|\$2) \rightarrow lr2.(lr1.(r_b.\$0|\$1)|\$2)$

$(r, exit, lr, \phi, lr) \Rightarrow lr.(r_b.\$0|\$1) \rightarrow r_b.\$0|lr.\$1$

$(r, open, o, lo, lr) \Rightarrow lr.(r_a.\$0|o_b.\$1|\$2)|\$3 \rightarrow lr.(r_x.\$0|o_x.\$1|\$2)|\$3$

$(r, close, o, lo, lr) \Rightarrow lr.(r_x.\$0|o_x.\$1|\$2)|\$3 \rightarrow lr.(r_a.\$0|o_b.\$1|\$2)|\$3$

$(r, login, o, lo, lr) \Rightarrow (1)lr.(r_a.\$0|o_b.\$1|\$2)|\$3 \rightarrow lr.(r_x.\$0|o_x.\$1|\$2)|\$3$

$(2)lr^*.(r_a.\$0|\$1)|lo^*.(o_b.\$2|\$3)|\$4 \rightarrow lr^*.(r_x.\$0|\$1)|lo^*.(o_x.\$2|\$3)|\$4$

$(r, logout, o, lo, lr) \Rightarrow (1)lr.(r_x.\$0|o_x.\$1|\$2)|\$3 \rightarrow lr.(r_a.\$0|o_b.\$1|\$2)|\$3$

$(2)lr^*.(r_x.\$0|\$1)|lo^*.(o_x.\$2|\$3)|\$4 \rightarrow lr^*.(r_a.\$0|\$1)|lo^*.(o_b.\$2|\$3)|\$4$

$(r, copy, o, lo, lr) \Rightarrow$

$(1)lr.(r_x.\$0|Loco1(lo)_x.(o|\$1)|\$2)|\$3 \rightarrow lr.(r_x.(o|\$0)|Loco1(lo)_x.(o|\$1)|\$2)|\$3$

$(2)lr^*.(r_x.\$0|\$1)|Loco2(lo)^*.(Loco1(lo)_x.(o|\$2)|\$3)|\$4 \rightarrow lr^*.(r_x.(o|\$0)|\$1)|Loco2(lo)^*.(Loco1(lo)_x.(o|\$2)|\$3)|\$4$

$(r, delete, o, lo, lr) \Rightarrow$

$(1)lr.(r_x.\$0|Loco1(lo)_x.(o|\$1)|\$2)|\$3 \rightarrow lr.(r_x.\$0|Loco1(lo)_x.\$1|\$2)|\$3$

$(2)lr^*.(r_x.\$0|\$1)|Loco2(lo)^*.(Loco1(lo)_x.(o|\$2)|\$3)|\$4 \rightarrow lr^*.(r_x.\$0|\$1)|Loco2(lo)^*.(Loco1(lo)_x.\$2|\$3)|\$4$

$(3)lr.(r_a.(o|\$0)|\$1)|\$2 \rightarrow lr.(r_a.\$0|\$1)|\$2$

eight operations and can be extended easily by administrators. As physical objects require that accessibility is the subject-object proximity, the authorization terms of the open and close operations are both mapped to one reaction rule. Hybrid and cyber objects can be accessed remotely, so the authorization terms of the login, logout, and copy operations are mapped to two reaction rules. When the objects and subjects are co-located, we obtain the first one. Otherwise, the second one is obtained. The delete action is mapped to three reaction rules because this action can also occur in the user's mobile phone. For the location of cyber objects, two functions are defined: $Loco2(lo)$, which returns the space location, and $Loco1(lo)$, which returns the hybrid object-location. The stars in lr^* and lo^* represent that they are at the same hierarchical level in bigraphs. For the bank example, the role lobby manager can copy file3 from the cloudlet in the main area. The redex is *mainarea*.(*lobbymanager*$_y$.\$1 | *corridor*.(*serverroom*.(*cloudlet*$_y$.*file*3 | \$2) | \$3)). The reactum is *mainarea*.(*lobbymanager*$_y$.(*file*3 | \$1) | *corridor*.(*serverroom*.(*cloudlet*$_y$.*file*3|\$2)|\$3)). In the redex and reactum, the server room is extended to the main area which is in the same hierarchy with the location of the lobby manager. The stars in lr^* and $Loco2(lo)^*$ have the same meaning.

Transformation rule 2 describes the transformation templates from authorization terms without topology constraints to reaction rules. When the authorization terms involve topology constraints, both the redex and reactum must satisfy these topology constraints.

Example 3 A security policy of the bank branch is "the client manager enters the president's office only when the president is in there". This requirement is expressed as (*clientmanager, enter, presidentoffice,* ϕ*, corridor, RoleLocation(president, presidentoffice)*. The

corresponding graphical bigraph is shown in Fig. 5.

Considering the enter transformation rule, we obtain the redex $corridor.(clientmanager_a.\$0 \mid presidentoffice.(president_b.\$1 \mid \$2) \mid \$3)$, and the reactum $corridor.(presidentoffice.(clientmanager_a.\$0 \mid president_b.\$1 \mid \$2) \mid \$3)$. The redex and reactum both satisfy the location constraint "the president is in the president's office".

Given a bigraph that describes the initial configuration, the system evolves by applying reaction rules, which model the occurrence of possible actions, generating new configurations. At each step, only the redex that matches the current configuration is selected. Several reaction rules may be possible for one configuration, thus branching off possible new configurations. Eventually, we obtain a Labeled Transition System (LTS).

4.4 Security requirements

A property for a given configuration can be expressed by bigraphs. A configuration described by a bigraph B satisfies a property D if the property D can be matched against B, exactly in the same way a redex is matched against a bigraph to apply a reaction rule. Failure to match the property D against B means that the property D is not satisfied in this configuration. The utilization of sites in the bigraph specifying the property that is checked against B indicates that the portion of B that matches a site does not affect the property. Checking properties only for a configuration is not enough; thus, we have to check system temporal requirements. Section 4.3 shows that the security policies are modeled as an LTS induced by reaction rules, so the modeling is conducted by predicting the branching structure using the branching Computation Temporal Logic (CTL). In CTL, two path quantifiers are A for "all paths" and E for "an existing path". Two common operators used to express state formula are G which is "in all states" and F which is "there exists some state".

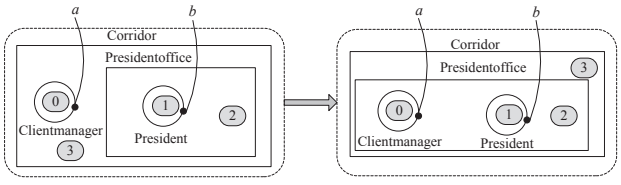


Fig. 5 Enter reaction rule.

We have introduced SoD constraints in Section 3.2.2, which cannot be violated. But beyond that, the correctness of relations defined in the TA-CPAC model also need to be checked. We present one example and its corresponding formal representation for each type of security requirement in Table 3.

4.5 Reduction approach

For the UE relation, the maximum number of explored states is $((C^1_{rolenum} + C^2_{rolenum} + \cdots + C^{rolenum}_{rolenum}) \times lrnum + 1)^{usernum}$, where $lrnum$, $rolenum$, and $usernum$ are the number of locations, roles, and users, respectively. Every user owns all roles and activates one role, two roles, and so on until all roles are activated in every location. This relationship is represented by combination formula. In addition, the user can deactivate all roles, so the formula adds one. This formula shows that the number of states increases exponentially with the number of users. For the PA relation, the maximum number of explored states is $lrnum^{rolenum} \times pernum^{rolenum}$, where $pernum$ is the number of permissions. We assume that every role can obtain all permissions in every location. The number of states increases exponentially with the number of roles.

Analyzing all explored states for a property has a negative effect on the analysis performance because some states and paths in the LTS model are irrelevant to the property to be verified. We first propose some definitions, which contribute to selecting the relevant elements of a property, and then provide an algorithm to select the related authorization terms based on these definitions.

Definition 8 Interactive relationship of users. If the topology constraints of the user u_j's UE relation

Table 3 TA-CPAC constraints and bigraphs representation.

Constraint	Bigraphs representation	Security requirement
OL	$AG\neg(lr.(o3_a.(o1 \mid o2 \mid \$0) \mid \$1)$	The cyber objects $o1$ and $o2$ cannot be co-located in the hybrid object $o3$ which is in the location lr.
UE	$EF(lr.(u_x.(r_x \mid \$0) \mid \$1) \mid \$2)$	The role r is enabled for the user u in the location lr.
PA	$EF(lr^*.(r_x \mid \$0) \mid lo^*.(o_x \mid \$1) \mid \$2)$	Eventually, the role r can log in the hybrid object o in the location lr ($lr \neq lo$).
RSSOD	$AG\neg(u_a.(r1_b \mid r2_c \mid \$0) \mid \$1)$	The roles $r1$ and $r2$ are conflicting roles.
LPSSOD	$AG\neg(lr^*.(r_y.\$0 \mid \$1) \mid lo1^*.(o1_y \mid \$2) \mid lo2^*.(o2_y \mid \$3) \mid \$4)$	The permissions logging in the hybrid objects $o1$ and $o2$ are conflicting permissions in the location lr .
LRSSOD	$AG\neg(lr.(u_x.(r1_x \mid r2_x \mid \$0) \mid \$1) \mid \$2)$	The roles $r1$ and $r2$ can not be both enabled in the location lr.

involve the user u_i's state or vice versa, then the users u_j and u_i have an interactive relationship.

Definition 9 Enclosed users. The enclosed users of a user u is a set that includes all the users who have interactive relationships with the user u in the UE relation.

Definition 10 Interactive relationship of roles. If the topology constraints of the role r_j's PA relation involve the role r_i's state or vice versa, then the roles r_j and r_i have an interactive relationship.

Definition 11 Enclosed roles. The enclosed roles of a role r is a set which includes all the roles who have interactive relationships with the role r in the PA relation.

Definition 12 Interactive relationship of locations. If the authorization terms in the location lr_i are constrained by the topological states of the location lr_j or vice versa, then the locations lr_j and lr_i have an interactive relationship.

Definition 13 Enclosed locations. The enclosed locations of a location lr is a set that includes all the locations, which have interactive relationships with the location lr in authorization terms.

Based on the concepts of enclosed users, enclosed roles, and enclosed locations, Algorithm 1 is proposed to select the property-related authorization terms. In this algorithm, we first extract the subjects and locations involved in the property. If the property relates with users, the $SP1$ set is selected based on the enclosed users. Otherwise, the $SP1$ set is selected based on the enclosed roles. Next, the subset $SP2$ is chosen from the set $SP1$ according to the enclosed locations.

Based on the definition of enclosed users, enclosed roles, and enclosed locations, the subjects and locations that are not in the enclosed sets are independent from the subjects and locations involved in the property. When the subjects $s1$ and $s2$ are independent, the behaviors of the subject $s1$ do not affect the subject $s2$ and vice versa. When the locations $lr1$ and $lr2$ are independent, the executable behaviors in area $lr1$ do not affect the behaviors in area $lr2$ and vice versa. Line 1 in Algorithm 1 aims to select the related subjects and locations. Lines 2–17 select the authorization terms based on the enclosed subjects, where lines 2–9 are based on the enclosed users and lines 11–17 are based on the enclosed roles. Lines 19–22 further select the related authorization terms based on the enclosed locations. Therefore, Algorithm 1 can ensure that unselected authorization terms do not contain elements that affect the property to be verified.

Algorithm 1 Selecting sub-authorization terms for a security property

Require: Property p, UE, PA;
Ensure: sub-authorization terms for property p;
1: $enSet = ExtractEntities(p)$;
/*get elements from the property, including subjects and locations*/
2: $U = GetUsers(enSet)$;
3: **if** $U != NULL$ **then**
4: $u = GetFirstElement(U)$;
5: **while** $u != NULL$ **do**
6: $cu = cu + GetEnclosedUsers(u, UE)$;
/*get enclosed users of the user u*/
7: $u = GetNextElement(U)$
8: **end while**
9: $SP1 = SearchSecurityPolicies1(cu, UE)$;
/*search user-role authorization terms based on enclosed users */
10: **else**
11: $R = GetRoles(enSet)$;
12: $r = GetFirstElement(R)$;
13: **while** $r != NULL$ **do**
14: $cr = cr + GetEnclosedRoles(r, PA)$;
/*obtain enclosed roles of the role r*/
15: $r = GetNextElement(R)$
16: **end while**
17: $SP1 = SearchSecurityPolicies2(cr, PA)$;
/*search role-permission authorization terms based on enclosed roles*/
18: **end if**
19: $LR1 = GetLocations(enSet)$;
20: $SP2 = SearchSecurityPolicies3(LR1, SP1)$;
/*search authorization terms based on locations involved in property p*/
21: $LR2 = GetEnclosedLocations(SP2)$;
/*obtain enclosed locations from topology constraints of $SP2$*/
22: $SP2 = SP2 + SearchSecurityPolicies3(LR2, SP1)$;
/*further search authorization terms based on enclosed locations*/
23: *return* $SP2$;

5 Evaluation

5.1 Case study

In this section, we illustrate the proposed model and verification method for a modern bank branch. We simplify this scenario to demonstrate the method introduced above. The deployment of the bank is described in Section 2. The initial topology configuration model and access control policies of the

bank are shown in Appendix A. We extract some important security requirements and check whether the security policies conform to them.

(1) The teller can own file2 in the accountant's office only when the president and accountant are both in this place.

(2) The president cannot enter the safe room when someone is in the president's office.

(3) The president cannot exit the president's office when file2 is on her mobile phone.

(4) The client manager cannot log in the server and cloudlet simultaneously in the client manager's office and main area.

(5) When the box is opened or closed by the teller, the accountant or president is in the teller's office.

(6) The teller and accountant are conflicting roles in the teller's office.

To support our approach, we created a prototype tool that transforms the topology configuration model and access control policies to the term language based on the transformation rules. The interface of the tool is shown in Fig. 6. The topological elements of the environment are input in the left part. These input parameters are formalized as the initial model. The access control policies are input in the right part. These parameters are formalized as reaction rules. Then, the initial model and reaction rules are organized and used as input to BigMC[29], which is a model checking tool for bigraphs and BRS, developed by Gian Perrone at IT University of Copenhagen. The verification is conducted on a Linux platform with 4 GB RAM using an Intel (R) Core i7-4710MQ 2.5 GHz processor.

The bigraphs representations of security requirements are presented in Table 4. The checking results are listed in Table 5, including enclosed subjects, enclosed locations, explored states, and required modeling time. We have proposed suggestions to solve the violated security properties in our previous work[30]. The final decision is obtained through discussion with the administrator.

Property 1 The property type of the first requirement is the PA relation with proximity constraints. The involved subjects in this property include the president, teller, and accountant. The involved location in this

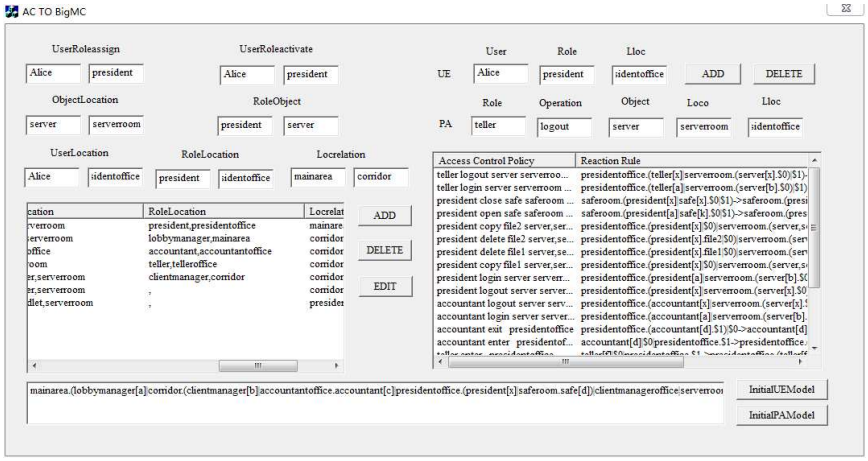


Fig. 6 Access control policies to BigMC language transformation tool.

Table 4 Bigraphs representations of properties.

Property	Bigraphs representation
Property 1	$AG\neg(president_a\|accountoffice.(teller_m.file2\|\$0) \wedge accountant_d\|accountoffice.(teller_m.file2\|\$0)\wedge$ $president_a\|accountant_d\|accountantoffice.(teller_m.file2\|\$0))$
Property 2	$AG\neg(presidentoffice.(saferoom.(president_a\|safe_b)\|accountant_d\|\$0)\|\$1\wedge$ $presidentoffice.(saferoom.(president_a\|safe_b)\|teller_f\|\$0)\|\$1\wedge$ $presidentoffice.(saferoom.(president_a\|safe_b)\|clientmanager_g\|\$0)\|\$1\wedge$ $presidentoffice.(saferoom.(president_a\|safe_b)\|lobbymanager_h\|\$0)\|\$1)$
Property 3	$EF(president_a.file2\|presidentoffice.\$0\|\$1)$
Property 4	$AG\neg(mainarea.(clientmanager_x.\$0\|corridor.(serverroom.(server_x.\$1\|cloudlet_x.\$2)\|\$3))\wedge$ $clientmanageroffice.clientmanager_x.\$0\|serverroom.(server_x.\$1\|cloudlet_x.\$2)\|\$3)$
Property 5	$AG\neg(telleroffice.(teller_x\|box_x)\|accountant_a\|president_b\|\$0)$
Property 6	$EF(telleroffice.(Tom_x.(teller_x\|accountant_x)))$

Table 5 Performance results.

Property	Enclosed subject	Enclosed location	Explored state	Time (s)
Property 1	president, accountant, teller	accountant office	190	0.195 029
Property 2	president, lobby manager, accountant, client manager, teller	president office, safe room	3890	8.866 34
Property 3	president, lobby manager, accountant, client manager, teller	president office	1514	2.779 99
Property 4	client manager	client manager office, main area, corridor, server room	41	0.027 646
Property 5	teller, accountant, president	teller office	356	0.319 504
Property 6	Tom	teller office	4	0.005 211

property is the accountant's office. The topology constraints in the $p23$ and $p27$ permissions of the teller, the $p24$ permission of the accountant, and the $p23$ permission of the president do not involve new elements. Therefore, the related permissions are in the *enabled_permission* (*president*, *accountantoffice*), *enabled_permission* (*accountant*, *accountantoffice*), and *enabled_permission* (*teller*, *accountantoffice*) sets. This property is violated. The counterexample path is shown in Fig. 7. To solve this problem, the $p24$ permission of the president and accountant is given the topology constraint $\neg$ *ObjectLocation* (*file2*, (*teller*, *accountantoffice*)).

Property 2 The property type of the second requirement is the PA relation with proximity constraints. The involved subjects of this property are the roles of president, accountant, teller, client manager, and lobby manager. The involved locations are the president's office and safe room. The topology constraints in the $p12$ permission of the president, and the $p11$ permission of the client manager, teller, accountant, and lobby manager do not involve new elements. Thus, the related permissions are in the *enabled_permission* (*president*, *presidentoffice*), *enabled_permission* (*accountant*, *presidentoffice*), *enabled_permission* (*clientmanager*, *presidentoffice*), *enabled_permission* (*lobbymanager*, *presidentoffice*), *enabled_permission* (*teller*, *presidentoffice*), and *enabled_permission* (*president*, *saferoom*) sets. This property is violated. To solve this problem, the $p19$ permission of the president is given the topology constraints $\neg(c11' \wedge c12' \wedge c13' \wedge c14')$.

Property 3 The property type of the third requirement is the PA relation with location constraints. The involved subject of this property is the role of president. The involved location is the president's office. The topology constraints in the $p12$ permission of the president are about the roles of accountant, client manager, lobby manager, teller, and the location of president's office. Therefore, the related permissions are in the *enabled_permission* (*president*, *presidentoffice*), *enabled_permission* (*accountant*, *presidentoffice*), *enabled_permission* (*clientmanager*, *presidentoffice*), *enabled_permission* (*lobbymanager*, *presidentoffice*), and *enabled_permission* (*teller*, *presidentoffice*) sets. This property is not violated.

Property 4 The property type of the fourth requirement is the PA relation with communication constraints. The involved subject of this property is the role of the client manager. The involved locations are the client manager's office, corridor, server room,

```
cc@cc-VirtualBox:~/Downloads/bigmc-master/doc/paper3/Evaluation/case study property checking$ ./bigmc  property1.bgm
*** Found violation of property: secure
*** secure: !matches(...)
#0  (accountantoffice.(teller[-].file2.nil | accountant[d].nil) | serverroom.(server[-].(file2.nil | file1.nil) | cloudlet[c].file3.nil) | president[a].nil)   <- *** VIOLATION ***
 >> president_exit_accountant

#1  (accountantoffice.(president[a].nil | teller[-].file2.nil | accountant[d].nil) | serverroom.(server[-].(file2.nil | file1.nil) | cloudlet[c].file3.nil))
 >> teller_copy_file2

#2  (accountantoffice.(teller[-].nil | accountant[d].nil | president[a].nil) | serverroom.(server[-].(file2.nil | file1.nil) | cloudlet[c].file3.nil))
 >> teller_login_server_presidentoffice

#3  (serverroom.(server[b].(file2.nil | file1.nil) | cloudlet[c].file3.nil) | accountantoffice.(accountant[d].nil | teller[f].nil | president[a].nil))
 >> teller_enter_accountant

#4  (accountantoffice.(accountant[d].nil | president[a].nil) | teller[f].nil | serverroom.(server[b].(file1.nil | file2.nil) | cloudlet[c].file3.nil))
 >> president_enter_accountant

#5  (accountantoffice.accountant[d].nil | president[a].nil | teller[f].nil | serverroom.(server[b].(file1.nil | file2.nil) | cloudlet[c].file3.nil))
 >> accountant_enter_accountant

#6  (president[a].nil | accountant[d].nil | teller[f].nil | accountantoffice.nil | serverroom.(server[b].(file1.nil | file2.nil) | cloudlet[c].file3.nil))
 >> (root)
time0.076894seconds
[mc::step] Counter-example found.
```

Fig. 7 Counterexample path.

and main area. The permissions of the role client manager in these locations do not involve topology constraints. Therefore, the related permissions are in the *enabled_permission* (*clientmanager*, *mainarea*), *enabled_permission* (*clientmanager*, *corridor*), *enabled_permission* (*clientmanager*, *clientmanageroffice*), and *enabled_permission* (*clientmanager*, *serverroom*) sets. This property is violated. To solve this problem, the $p37$ permission of the client manager is given the topology constraints $\neg$(*RoleObject* (*clientmanager*, *cloudlet*) $\wedge$ *RoleLocation* (*clientmanager*, *clientmanageroffice*) $\wedge$ *ObjectLocation* (*cloudlet*, *serverroom*)). The $p41$ permission of the client manager is given the topology constraints $\neg$ (*RoleObject* (*clientmanager*, *server*) $\wedge$ *RoleLocation* (*clientmanager*, *clientmanageroffice*) $\wedge$ *ObjectLocation* (*server*, *serverroom*)).

Property 5 The property type of the fifth requirement is the PA relation with proximity constraints. The involved subjects of this property are the roles of the teller, accountant, and president. The involved location is the teller's office. The topology constraint $c10'$ in permissions of the accountant and president is about the role of the teller and the location of the teller's office. Therefore, the related permissions are in the *enabled_permission* (*president*, *telleroffice*), *enabled_permission* (*accountant*, *telleroffice*), and *enabled_permission* (*teller*, *telleroffice*) sets. This property is violated. To solve this problem, the $p46$ and $p47$ permissions of the teller are given the topology constraints *RoleLocation* (*accountant*, *telleroffice*) || *RoleLocation* (*president*, *telleroffice*).

Property 6 The property type of the sixth requirement is the LRSSOD. Only Tom is assigned the roles teller and accountant. The involved subject of this property is Tom. The involved location is the teller's office. Therefore, the related roles are in the *enabled_role* (*Tom*, *telleroffice*) set. This property is violated. The UE relation (*Tom*, *accountant*, *telleroffice*) is rescinded.

When the UE and PA relations are modified, we only need to re-verify a part of properties in which the related authorization terms are changed.

5.2 Performance analysis

In this section, the performance of the policy verification method is evaluated by varying the number of users, roles, and locations. Although hundreds or thousands of subjects exist in an organization, only the enclosed subjects and locations are involved in a verified property in our method. The maximum number of enclosed users for the UE relation is set to five. The maximum number of enclosed roles for the PA relation is set to five. The maximum number of enclosed locations is set to six. These parameters were selected to match those used in Ref. [22].

In Fig. 8, we vary the number of enclosed users, considering that the number of enclosed roles and the number of enclosed locations are both two. x-axis represents the number of enclosed users and y-axis represents the number of states explored and time required. Figure 9 shows the number of explored states and time required by varying the number of enclosed roles, considering that the number of locations is two and the number of permissions is four. These two figures show that explored states and the execution time increase exponentially with the increase in the number of users and roles. When the number of users increases, the number of states combination of users increases significantly. For example, two roles are assigned to each user. The user can activate two roles, one role, and zero role. The number of states of a user is four. If five users are involved, then the number of states is 1024. When the number of roles increases, the sequence of actions among them becomes complicated. For example, three subjects enter a location. The behavior sequence among them is not constrained. The number

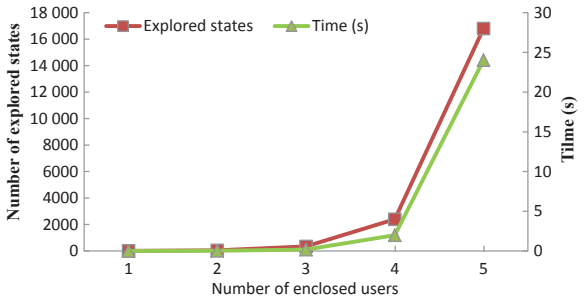


Fig. 8 Number of explored states and the time for the UE relation with varying number of enclosed users.

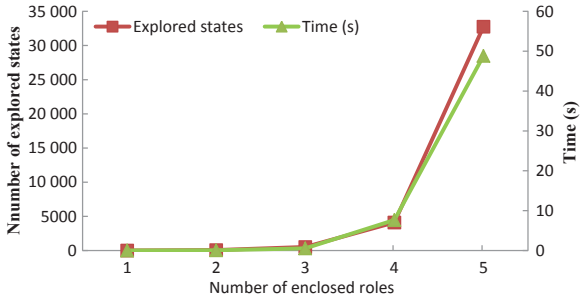


Fig. 9 Number of explored states and the time for the PA relation with varying number of enclosed roles.

of states is eight. If ten subjects enter a location, then the number of states is 1024. Thus, the number of subjects is the most effective factor in the number of explored states and required time. As shown in Fig. 10, we vary the number of enclosed locations, considering that the number of enclosed users and number of enclosed roles are both two. x-axis represents the number of enclosed locations and y-axis represents the number of states explored and required time. Figure 11 is used to analyze the PA relations by varying the number of enclosed locations. The numbers of roles and permissions are two and four, respectively. These two figures indicate that the number of locations is proportional to the explored states and required time in UE relation and PA relation.

These experimental results are consistent with our theoretical analysis introduced in Section 4.5 and also show the effectiveness of our reduction algorithm. Meanwhile, this conclusion can guide the generation of policies that facilitate security analysis.

6 Conclusion and Future Work

Little work has been done on the cyber-physical access control field and existing ones do not consider the security of the cyber and physical worlds simultaneously. For the CPP, a unified access control model is necessary to ensure that cyber and physical resources are both handled securely regardless of how the operational environment changes. To meet these requirements, this study proposes a TA-CPAC model and an approach for engineering security policies. We first present the topology configuration model, which can provide valuable contextual indicators for the access decision. Based on this model, the TA-CPAC model is presented, which integrates the physical access control and cyber access control, and can ensure the security of the cyber and physical spaces at the same time. Then, we use bigraphs and BRS to represent and rationalize our proposed security policies to ensure their conformity with security requirements. A reduction algorithm is also proposed to simplify the reasoning process. Finally, we demonstrate the applicability of our approach through a case study. Our findings are encouraging and provide evidence of the feasibility of the approach.

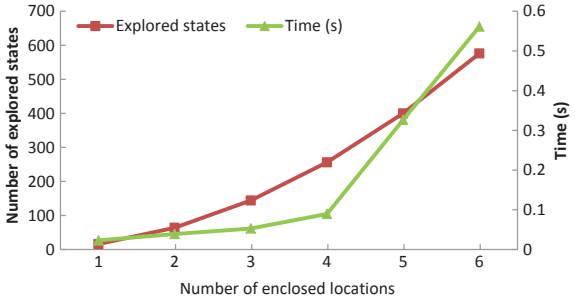


Fig. 10 Number of explored states and the time for the UE relation with varying number of enclosed locations.

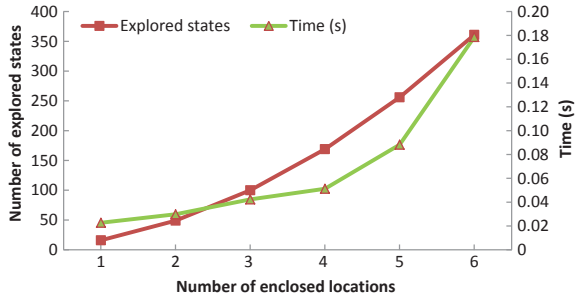


Fig. 11 Number of explored states and the time for the PA relation with varying number of enclosed locations.

We have identified and are pursuing a number of promising avenues for further investigations. This paper only considers the access control for a single administrative domain. Multiple domains may need to cooperate with each other to achieve common goals. Our next work aims to focus on how to realize secure cross-domain operations in collaborative cyber-physical spaces. In emergency situations, access levels might need to be temporarily downgraded or reconfigured to bring the system under control. We will also study how to manage the risk scenario in the CPP. We aim to provide a complete policy framework for the CPP with an integrated specification, verification, and enforcement environment.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61772270, 61602262, and 61602237), Jiangsu Natural Science Foundation of China (No. BK20170809), the National High-Tech Research and Development (863) Program of China (No. 2015AA015303), and Science Foundation of Nanjing Institute of Technology (No. YKJ201420).

References

[1] C. Tsigkanos, L. Pasquale, C. Ghezzi, and B. Nuseibeh, On the interplay between cyber and physical spaces for adaptive security, *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 3, pp. 466–480, 2018.

[2] D. Chen, G. R. Chang, D. W. Sun, J. Jia, and X. W. Wang, Modeling access control for cyber-physical systems using

reputation, *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1088–1101, 2012.

[3] M. Toahchoodee and I. Ray, On the formalization and analysis of a spatio-temporal role-based access control model, *J. Comput. Secur.*, vol. 19, no. 3, pp. 399–452, 2011.

[4] M. S. Kirkpatrick, M. L. Damiani, and E. Bertino, Prox-RBAC: A proximity-based spatially aware RBAC, in *Proc. 19th ACM Int. Conf. on Advances in Geographic Information Systems*, Chicago, IL, USA, 2011, pp. 339–348.

[5] A. Gupta, M. S. Kirkpatrick, and E. Bertino, A formal proximity model for RBAC systems, *Comput. Secur.*, vol. 41, pp. 52–67, 2014.

[6] A. B. Fadhel, D. Bianculli, and L. Briand, A comprehensive modeling framework for role-based access control policies, *J. Syst. Soft.*, doi: 10.1016/j.jss.2015.05.015.

[7] J. W. Huang, D. M. Nicol, R. Bobba, and J. H. Huh, A framework integrating attribute-based policies into role-based access control, in *Proc. 17th ACM Symp. Access Control Models and Technologies*, New York, NY, USA, 2012, pp. 187–196.

[8] X. Jin, R. Sandhu, and R. Krishnan, RABAC: Role-centric attribute-based access control, in *Proc. 6th Int. Conf. Mathematical Methods, Models, and Architectures for Computer Network Security*, St. Petersburg, Russia, 2012, pp. 84–96.

[9] N. Skandhakumar, F. Salim, J. Reid, F. Jason, and E. Dawson, Physical access control administration using building information models, presented at the 4th Int. Conf. Cyberspace Safety and Security, Melbourne, Australia, 2012.

[10] F. Turkmen, S. Foley, B. O'Sullivan, W. Fitzgerald, T. Hadzic, S. Basagiannis, and M. Boubekeur, Explanations and relaxations for policy conflicts in physical access control, presented at the 25th Int. Conf. Tools with Artificial Intelligence, Herndon, VA, USA, 2014.

[11] E. Geepalla, B. Bordbar, and X. F. Du, Spatio-temporal role based access control for physical access control systems, presented at the 14th Int. Conf. Emerging Security Technologies, Cambridge, UK, 2013.

[12] C. Tsigkanos, T. Kehrer, and C. Ghezzi, Modeling and verification of evolving cyber-physical spaces, in *Proc. 11th Joint Meeting on Foundations of Software Engineering*, New York, NY, USA, 2017, pp. 38–48.

[13] I. Ray and I. Ray, Access control challenges for cyber-physical systems, http://pdfs.semanticscholar.org/953a/5ef6ae9a1983469a0393f525f76d33191f94.pdf, 2009.

[14] V. C. Hu and R. Kuhn, Access control policy verification, *Computer*, vol. 49, no. 12, pp. 80–83, 2016.

[15] X. Zhao and M. E. Johnson, Access governance: Flexibility with escalation and audit, in *Proc. 43rd Hawaii Int. Conf. System Sciences*, Washington, DC, USA, 2010.

[16] V. C. Hu, D. R. Kuhn, T. Xie, and J. Hwang, Model checking for verification of mandatory access control models and properties, *Int. J. Softw. Eng. Know. Eng.*, vol. 21, no. 1, pp. 103–127, 2011.

[17] D. Unal and M. U. Caglayan, Spatiotemporal model checking of location and mobility related security policy specifications, *Turk. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 144–173, 2013.

[18] D. Unal and M. U. Caglayan, A formal role-based access control model for security policies in multi-domain mobile networks, *Comput. Networks*, vol. 57, no. 1, pp. 330–350, 2013.

[19] S. Jha, N. H. Li, M. Tripunitara, Q. H. Wang, and W. Winsborough, Towards formal verification of role-based access control policies, *IEEE Trans. Depend. Secure Comput.*, vol. 5, no. 4, pp. 242–255, 2008.

[20] A. Gouglidis, I. Mavridis, and V. C. Hu, Security policy verification for multi-domains in cloud systems, *Int. J. Inf. Secur.*, vol. 13, no. 2, pp. 97–111, 2014.

[21] M. Toahchoodee, I. Ray, K. Anastasakis, G. Georg, and B. Bordbar, Ensuring spatio-temporal access control for real-world applications, in *Proc. 14th ACM Symp. Access Control Models and Technologies*, New York, NY, USA, 2009, pp. 13–22.

[22] S. Mondal, S. Sural, and V. Atluri, Security analysis of GTRBAC and its variants using model checking, *Comput. Secur.*, vol. 30, nos. 2&3, pp. 128–147, 2011.

[23] L. Pasquale, C. Ghezzi, E. Pasi, C. Tsigkanos, M. Boubekeur, B. Florentino-Liano, T. Hadzic, and B. Nuseibeh, Topology-aware access control of smart spaces, *Computer*, vol. 50, no. 7, pp. 54–63, 2017.

[24] L. A. Walton and M. Worboys, A qualitative bigraph model for indoor space, presented at the 6th Int. Conf. Geographic Information Science, Berlin, Germany, 2012.

[25] C. Tsigkanos, T. Kehrer, and C. Ghezzi, Architecting dynamic cyber-physical spaces, *Computing*, vol. 98, no. 10, pp. 1011–1040, 2016.

[26] S. Benford, M. Calder, T. Rodden, and M. Sevegnani, On lions, impala, and bigraphs: Modelling interactions in physical/virtual spaces, *ACM Trans. Comput.-Human Interact.*, vol. 23, no. 2, pp. 1–56, 2016.

[27] L. Pasquale, C. Ghezzi, C. Menghi, C. Menghi, and B. Nuseibeh, Topology aware adaptive security, in *Proc. 9th Int. Symp Software Engineering for Adaptive and Self-Managing Systems*, New York, NY, USA, 2014, pp. 43–48.

[28] R. Milner, *The Space and Motion of Communicating Agents*. Cambridge, UK: Cambridge University Press, 2009.

[29] G. Perrone, S. Debois, and T. T. Hildebrandt, A verification environment for bigraphs, *Innovat. Syst. Soft. Eng.*, vol. 9, no. 2, pp. 95–104, 2013.

[30] Y. Cao, Z. Q. Huang, S. L. Kan, H. F. Peng, and C. B. Ke, Location-constrainted access control model and verification methods, (in Chinese), *J. Comput. Res. Dev.*, vol. 55, no. 8, pp. 1809–1825, 2018.

Appendix A: Topology configuration model and TA-CPAC model of the bank branch.

Table A1 Topology configuration model of Fig. 1.

Set	Value
User	{Alice, Bob, Tom, Jone, Clark}
Role	{president, lobbymanager, accountant, teller, clientmanager}
Object	{server, cloudlet, box, safe, file1, file2, file3, Alicephone, Bobphone, Tomphone, Jonephone, Clarkphone}
Lloc	{saferoom, accountantoffice, serverroom, presidentoffice, clientmanageroffice, telleroffice, mainarea, corridor}
Loco	{saferoom, accountantoffice, serverroom, presidentoffice, clientmanageroffice, telleroffice, mainarea, corridor, server, cloudlet, Alicephone, Bobphone, Tomphone, Jonephone, Clarkphone}
UserRoleassign	{(Alice, president), (Bob, lobbymanager), (Clark, accountant), (Tom, accountant), (Tom, teller), (Jone, teller), (Jone, clientmanager)}
UserRoleactivate	{(Alice, president), (Jone, clientmanager), (Bob, lobbymanager), (Tom, teller), (Clark, accountant)}
UserLocation	{(Alice, presidentoffice), (Bob, mainarea), (Clark, accountantoffice), (Tom, telleroffice), (Jone, corridor)}
RoleLocation	{(president, presidentoffice), (lobbymanager, mainarea), (accountant, accountantoffice), (teller, telleroffice), (clientmanager, corridor)}
ObjectLocation	{(server, serverroom), (cloudlet, serverroom), (box, telleroffice), (safe, saferoom), (file1, server, serverroom), (file2, server, serverroom), (file3, cloudlet, serverroom), (Alicephone, presidentoffice), (Bobphone, mainarea), (Clarkphone, accountantoffice), (Tomphone, telleroffice), (Jonephone, corridor)}
Locrelation	{(mainarea, corridor), (corridor, serverroom), (corridor, telleroffice), (corridor, presidentoffice), (corridor, clientmanageroffice), (corridor, accountantoffice), (presidentoffice, saferoom)}
RoleObject	{(president, server)}

Table A2 UE relation with topology constraints.

User	Role	Location and topology constraint
Alice	president	corridor, teller office, president office, safe room, main area, accountant office
Bob	lobby manager	corridor, main area, president office
Clark	accountant	main area, corridor, accountant office, teller office, president office
Tom	accountant	main area, corridor, accountant office $c1$, teller office, president office
Tom	teller	main area, client manager office, corridor, accountant office $c2$, teller office, president office
Jone	teller	main area, corridor, client manager office, president office, teller office, accountant office
Jone	client manager	corridor, client manager office, president office, mainarea

Table A3 PA relation with topology constraints.

Role	Permission and topology constraint
President	$p1-p6$, $p7\ c10'$, $p8\ c10'$, $p9-p11$, $p12\ \neg(c11'\wedge c12'\wedge c13'\wedge c14'\wedge c15')$ $p13-p22$, $p23\ c7'$, $p24-p30$
Clientmanager	$p9$, $p10$, $p11\ c6'$, $p12-p14$, $p18$, $p31-p45$
Teller	$p1-p4$, $p6$, $p9$, $p10$, $p11\ c6'$, $p12-p14$, $p18$, $p23\ c7'$, $p24-p26$, $p27\ c8'$, $p30$, $p35$, $p36$, $p46-p51$
Accounant	$p1-p4$, $p6$, $p9$, $p10$, $p11\ c6'$, $p12-p14$, $p18$, $p23$, $p24\ \neg(c8'\wedge c9')$, $p25$, $p26$, $p27\ c8'$, $p30$, $p46\ c10'$, $p47\ c10'$, $p52-p55$
Lobbymanager	$p9$, $p10$, $p11\ c6'$, $p12-p14$, $p18$, $p31\ \neg(c4'\wedge c5'\wedge c3')$, $p32$, $p34$, $p56\ \neg(c1'\wedge c2'\wedge c3')$, $p57-p59$

Table A4 Permissions of the bank.

No.	Permission
$p1$	enter telleroffice (ϕ) corridor
$p2$	exit telleroffice (ϕ) telleroffice (enter corridor (ϕ) telleroffice)
$p3$	login server (server, serverroom) telleroffice
$p4$	copy file1 (server, serverroom) telleroffice
$p5$	delete file1 (president, telleroffice) telleroffice
$p6$	logout server (server, serveroom) telleroffice
$p7$	open box (telleroffice) telleroffice
$p8$	close box (telleroffice) telleroffice
$p9$	enter corridor (ϕ) mainarea
$p10$	enter mainarea (ϕ) corridor
$p11$	enter presidentoffice (ϕ) corridor
$p12$	enter corridor (ϕ) presidentoffice
$p13$	login server (serverroom) presidentoffice
$p14$	copy file1 (server, serverroom) presidentoffice
$p15$	copy file2 (server, serverroom) presidentoffice
$p16$	delete file1 (president, presidentoffice) presidentoffice
$p17$	delete file2 (president, presidentoffice) presidentoffice
$p18$	logout server (serverroom) presidentoffice
$p19$	enter saferoom (ϕ) presidentoffice
$p20$	open safe (saferoom) saferoom
$p21$	close safe (saferoom) saferoom
$p22$	enter presidentffice (ϕ) saferoom
$p23$	enter accountantoffice (ϕ) corridor
$p24$	enter corridor (ϕ) accountantoffice
$p25$	login server (serverroom) accountantoffice
$p26$	copy file1 (server, serverroom) accountantoffice
$p27$	copy file2 (server, serverroom) accountantoffice
$p28$	delete file1 (president, accountantoffice) accountantoffice
$p29$	delete file2 (president, accountantoffice) accountantoffice
$p30$	logout server (serverroom) accountantoffice
$p31$	login cloudlet (serverroom) mainarea
$p32$	copy file3 (server, serverroom) mainarea
$p33$	delete file3 (clientmanager, mainarea) mainarea
$p34$	logout cloudlet (serverroom) mainarea
$p35$	enter clientmanageroffice (ϕ) corridor
$p36$	enter corridor (ϕ) clientmanageroffice
$p37$	login server (serverroom) clientmanageroffice
$p38$	copy file1 (server, serverroom) clientmanageroffice
$p39$	delete file1 (clientmanager, clientmanager) clientmanageroffice
$p40$	logout server (serverroom) clientmanageroffice
$p41$	login cloudlet (serverroom) clientmanageroffice
$p42$	copy file3 (server, serverroom) clientmanageroffice
$p43$	delete file3 (clientmanager, clientmanageroffice) clientmanageroffice
$p44$	logout cloudlet (serverroom) clientmanageroffice
$p45$	delete file1 (clientmanager, presidentoffice) presidentoffice

(To be continued)

Table A4 Permissions of the bank. (Continued)

No.	Permission
$p46$	open box (telleroffice) telleroffice
$p47$	close box (telleroffice) telleroffice
$p48$	delete file1 (teller, telleroffice) telleroffice
$p49$	delete file1 (teller, presidentoffice) presidentoffice
$p50$	delete file1 (teller, accountanoffice) accountantoffice
$p51$	delete file2 (teller, accountanoffice) accountantoffice
$p52$	delete file1 (accountan, telleroffice) telleroffice
$p53$	delete file1 (accountant, presidentoffice) presidentoffice
$p54$	delete file1 (accountant, accountantoffice) accountantoffice
$p55$	delete file2 (accountant, accountantoffice) accountantoffice
$p56$	login server (serverroom) mainarea
$p57$	logout server (serverroom) mainarea
$p58$	delete file3 (lobbymanager, mainarea) mainarea
$p59$	delete file1 (lobbymanager, presidentoffice) presidentoffice

Table A5 Topology constraints on the UE relation.

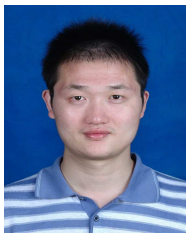
No.	Topology constraint
$c1$	$\neg(UserRoleactivate\ (Tom, teller) \wedge UserLocation\ (Tom, accountantoffice))$
$c2$	$\neg(UserRoleactivate\ (Tom, accountant) \wedge UserLocation\ (Tom, accountantoffice))$

Table A6 Topology constraints on the PA relation.

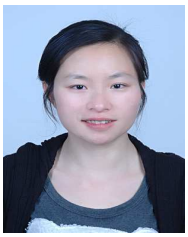
No.	Topology constraint
$c1'$	$RoleObject\ (lobbymanager, clouldet)$
$c2'$	$ObjectLocation\ (cloudet, serverroom)$
$c3'$	$RoleLocation\ (lobbymanager, mainarea)$
$c4'$	$RoleObject\ (lobbymanager, server)$
$c5'$	$ObjectLocation\ (server, serverroom)$
$c6'$	$RoleLocation\ (president, presidentoffice)$
$c7'$	$RoleLocation\ (accountant, accountantoffice)$
$c8'$	$RoleLocation\ (president, accountantoffice)$
$c9'$	$RoleLocation\ (teller, accountantoffice)$
$c10'$	$RoleLocation\ (teller, telleroffice)$
$c11'$	$RoleLocation\ (clientmanager, presidentoffice)$
$c12'$	$RoleLocation\ (teller, presidentoffice)$
$c13'$	$RoleLocation\ (lobbymanager, presidentoffice)$
$c14'$	$RoleLocation\ (accountant, presidentoffice)$
$c15'$	$ObjectLocation\ (file2, (president, presidentoffice))$

Yan Cao received the MS degree from Zhengzhou University in 2010. She is currently a PhD candidate in the Department of Computer Sciences and Technology, Nanjing University of Aeronautics and Astronautics. Her research interests include smart space, information security, formal methods, and access control.

Zhiqiu Huang received the PhD degree from Nanjing University of Aeronautics and Astronautics in 1999. He is currently a professor at the Department of Computer Sciences and Technology, Nanjing University of Aeronautics and Astronautics. He has supported several research projects of the National Natural Science Foundation of China about privacy and security. He has participated in several industrial projects working on the area of IT security. His research interests include safety and dependable software development, formal methods, service-oriented architecture, and security protocols.



Shuanglong Kan received the PhD degree from Nanjing University of Aeronautics and Astronautics in 2017. He is currently a post-doctoral researcher in Nanjing University of Aeronautics and Astronautics. His research interests include model checking, theorem proving, and refinement-based software development.



Dajuan Fan received the PhD degree from Nanjing University of Aeronautics and Astronautics in 2015. She is currently a post-doctoral researcher in Nanjing University of Aeronautics and Astronautics and also a teacher at the Department of Computer Sciences and Technology, Nanjing Institute of Technology. Her research interests include web service, software engineering, privacy protection, and formal methods.

Yang Yang received the bachelor degree from Nanjing University of Aeronautics and Astronautics in 2016. He is currently a master student at Nanjing University of Aeronautics and Astronautics. His research interests include access control policies, social network, and formal methods.