IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 17, 2017, accepted April 4, 2017, date of publication April 19, 2017, date of current version June 7, 2017.

*Digital Object Identifier 10.1109/ACCESS.2017.2693380*

# An Access Control Model for Resource Sharing Based on the Role-Based Access Control Intended for Multi-Domain Manufacturing Internet of Things

**QIANG LIU[1], HAO ZHANG[1], JIAFU WAN[2], (Member, IEEE), AND XIN CHEN[1]**

[1]Key Laboratory of Computer Integrated Manufacturing System, Guangdong University of Technology, Guangzhou 510006, China
[2]School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510006, China

Corresponding author: Xin Chen (xchen@gdut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51675108, in part by the Science and Technology Planning Project of Guangdong Province of China under Grant 2015B010128007 and Grant 2016A010106006, and in part by the Fundamental Research Funds for the Central Universities under Grant 2015ZZ079.

**ABSTRACT** Manufacturing Internet of Things (MIoT) represents the manufacturing oriented to Internet of Things with two important characteristics, resource sharing and process collaboration. Access control in resource sharing is very important for MIoT operation safety. This paper presents an access control model for resource sharing based on the role-based access control intended for multidomain MIoT. In multidomain systems, in order to response on the assigning request for permission for the certain role from the certain user, an authority action sequence named the authorization route is employed to determine an appropriate authorization state. In this paper, the best authorization route with the least spread of permissions is defined as an optimal authorization route. We employed an intelligent planning theory to model the authorization route problem and to develop a solution algorithm called PGAO*, which can support external evaluation of both single-goal-role authorization routes and multi-goal-role authorization routes. In addition, some simple policies for solving the authorization route problem are presented. The proposed access control model provides a quick and efficient authorization decision support for administrators in collaborative domain and ensures a secure access in resource sharing in MIoT.

**INDEX TERMS** Access control model, authorization route, manufacturing internet of things, role-based access control.

## I. INTRODUCTION

The Internet to Things (IoT) has brought a new revolution in manufacturing, consumption and logistic processes, and human everyday life. Namely, IoT can send information to different targets, share information among multiple targets, and achieve the connection ''things to things'' through comprehensive perception, reliable transmission, and intelligent processing. RFID is a major prerequisite for IoT, which connects physical objects through the Internet [1]. The Manufacturing Internet of Things (MIoT) represents an in-depth integration of manufacturing and IoT. More specifically, MIoT closely links organizations, resources, information, objects, and people through standard protocols [2], information sensing devices, and heterogeneous networks [3], [4]. This action enables the fabrication of products and services in the network where they can identify each other. MIoT transforms the decentralized factories into a unified intelligent manufacturing environment. In addition, MIoT forms an intelligent network composed of ubiquitous sensors, embedded terminal systems, intelligent control systems, and communication facilities through Cyber-Physical Systems (CPSs). The CPSs have emerged as a cutting edge technology for next generation industrial applications, and are undergoing rapid development and inspiring numerous application domains [5]. These emerging technology brings great opportunities for promotion of industrial upgrades and even allow the introduction of the fourth industrial revolution, namely, Industry 4.0 [6]. The smart factory is an important feature of Industry 4.0 that addresses vertical integration and networks the manufacturing systems for smart production,

therefore, the framework and operational mechanism of smart factory were introduced in [7]. The network interconnections between people, people and machines, and machines and machines, as well as services, allow horizontal, vertical, and end-to-end integration of intra- and inter-enterprise, and the entire value chain.

The main features of MIoT are resource sharing and process collaboration. Resource sharing is managed such that all production resources can be easily accessed in networks that have unified mechanical, electrical and communications standards, which enables the worldwide production resource configurations. On the other hand, process collaboration provides manufacturing of resources, information, and products. Therefore, the human resources can collaborate and complete manufacturing and design tasks in MIoT closely and orderly [8]. MIoT consists of many cross-business, cross-organization and cross-region resources than exchange information, thus, the security problems of Internet also refers to MIoT [9]. Moreover, the sharing extent is much higher than in the previous information systems. Therefore, it is necessary to ensure that information stays confidential, integral, and undeniable during collection, transmission, processing and accessing. In addition, access control and authorization optimization are very important. Lastly, the information leakage, tampering and minimal permission spread are also required. The mentioned parameters represent the crucial issues of designing of MIoT collaborative work environment.

Nowadays, the Role-Based Access Control (RBAC) model [11], [12] is the most popular access control model for collaborative environments, which is widely used for access control of resources and objects. In RBAC model, permissions are associated with roles and users are considered as members of corresponding roles. Roles, which are intermediary layers between permissions and users, can simplify the assigning and revoking of permissions, and can support an automatic authorization to certain extent. The administrative RBAC model, ARBAC97 [12], contains a self-management mechanism and a distributed authorization mechanism to satisfy large-scale access control requirements. The distributed authorization management mechanism of RBAC model is ideal for a multi-domain interoperation of access control. In [13] and [14], the OS-RBAC was designed based on RBAC principle, and the organization architecture and management boundaries were proposed and defined clearly.

In this study, RBAC is employed for resource access control for collaborative process in MIoT. Namely, we set up a formal model for RBAC safety policies, defined the Authorization Route Optimizing Problem (AROP) and designed a solution algorithm, named PGAO∗, which integrates the graph-planning algorithm and the AO star algorithm. The contributions of this work are as follows. We propose a resource access control model for resources sharing in MIoT, define AROP, and design a solution algorithm, which could decrease the security administrator workloads. We believe that, once the AROP is calculated automatically using the corresponding RBAC management mechanism, an automatic authorization can be implemented.

The paper is organized as follows. In Section 2, the related work is presented. In Section 3, AROP is formally defined. The proposed solution algorithm PGAO* is studied in Section 4. The evaluation of authorization route is introduced in Section 5. The multi-goal-role AROPs are presented in Section 6. The computational complexity and some simple policies for AROP are explained in Section 7. Finally, a brief conclusion is given in Section 8.

## II. RELATED WORK

The AROP can provide decision support for administrators when they execute authorization according to security policies and rules. However, there are many similar problems that refer to safety analysis [15], simple safety [16]–[17], security analysis [18], reachability [19], and user-role reachability [20], which are discussed in the following.

### A. SAFETY ANALYSIS

The safety analysis shows whether a command leaks a generic right from current configuration to future configuration. Here, the command refers to authorization and configuration, and corresponds to RBAC authorization state.

### B. SIMPLE SAFETY

The simple safety checks whether there is a reachable state in which a specific (presumably un-trusted) principal has access to a given resource. It was firstly proposed for trust management as one of security analyses [16].

### C. SECURITY ANALYSIS

The security analysis checks whether there is a state wherein principals from un-trusted principal set can change current state to goal state in which an access request is permitted. This represents a PSPACE-complete problem and factors that contribute to the computational complexity studied by consideration of various sub cases of this problem using different restrictions [18].

### D. REACHABILITY

The reachability determines whether the user has a given role in any policy that is reachable from initial policy via actions by a given set of administrators. This is also a PSPACE-complete problem [19].

### E. USER-ROLE REACHABILITY

The user-role reachability focuses on the possibility that user is assigned to roles by administrators. This problem is intractable, thus, the parameterized complexity is analyzed in [20] based on the classes of policies.

The listed problems have problem structure similar to ARPP. Namely, this paradigms seek a goal state in which user that corresponds to principal or subject can access to resource that corresponds to object. The difference is that above-mentioned problems represent the decision problems,

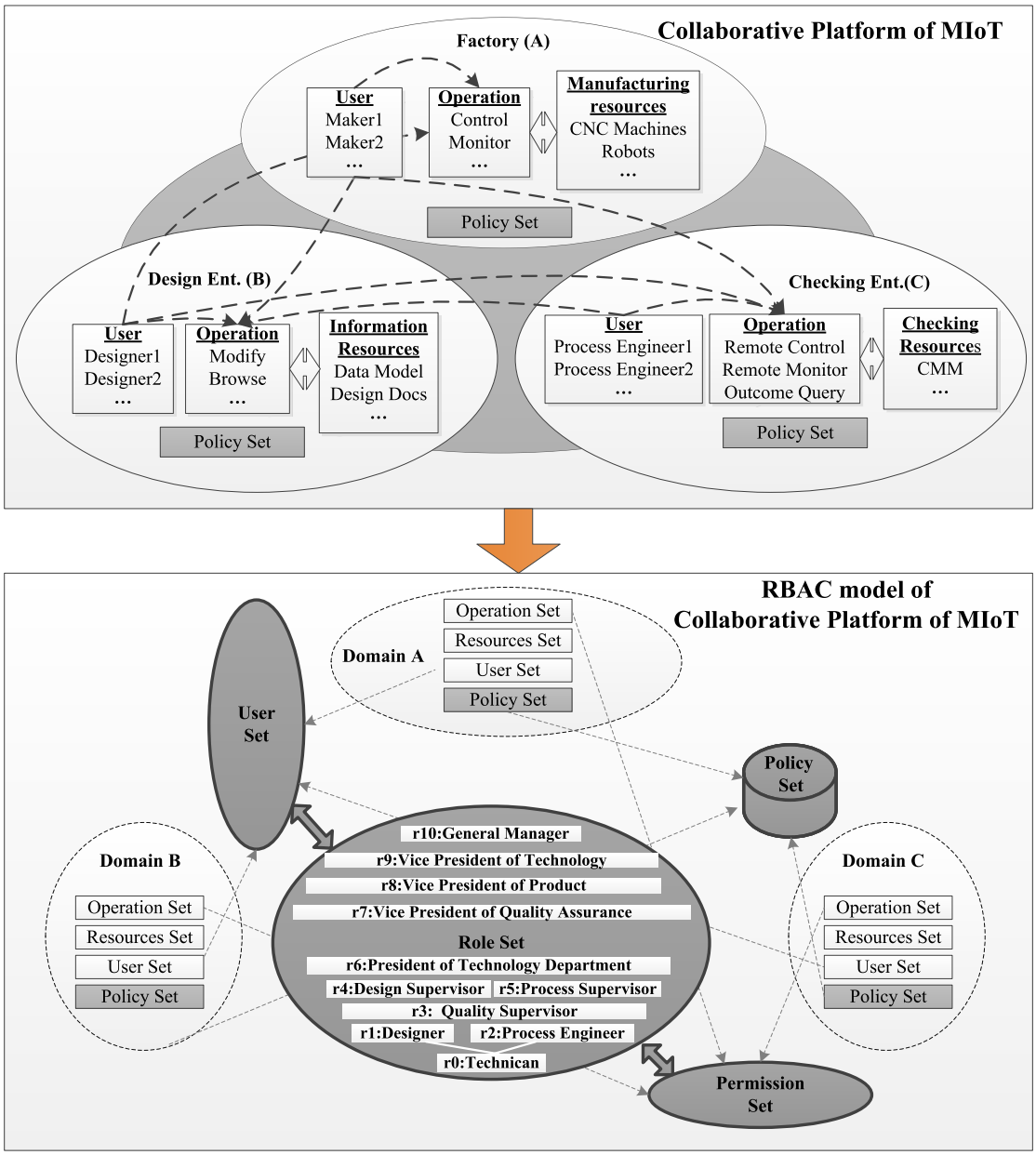



**FIGURE 1.** Access control in the collaborative platform of MIoT.

while ARPP represents a planning problem. However, AROP is an optimization problem. Besides AROP tries to find the authorization action and optimal authorization action route. All of these paradigms, including AROP, provide decision support to administrators and determine whether a state or an authorization action route exists. Moreover, AROP estimates the best authorization action route to obtain the certain state. The reachability problem, presented in [20], was initially modeled as a planning problem, which has guided our works greatly.

The AROP of RBAC model is a policy-related problem that is based on assumption that the policy set, on which AROP depends, is consistent. Various methods have been proposed to check the policy consistency, including standard deontic logic [21], model checking theory and tools [22], description logic [23] and the logic programming approach [24]. However, different checking approaches are based on different expressive languages. The policy consistency problem is more like a static problem of RBAC model. In contrast, AROP is a typical runtime problem of RBAC model. The difference between these problems is clear. Namely, the policy consistency problem focuses on the semantic conflicts among policy sets, while ARP focuses on the reasoning relationships among policy sets. Conversely, there are significant relationships between these problems. Particularly, before attempting to solve the AROP, we always assume that policy set related to the AROP is consistent.

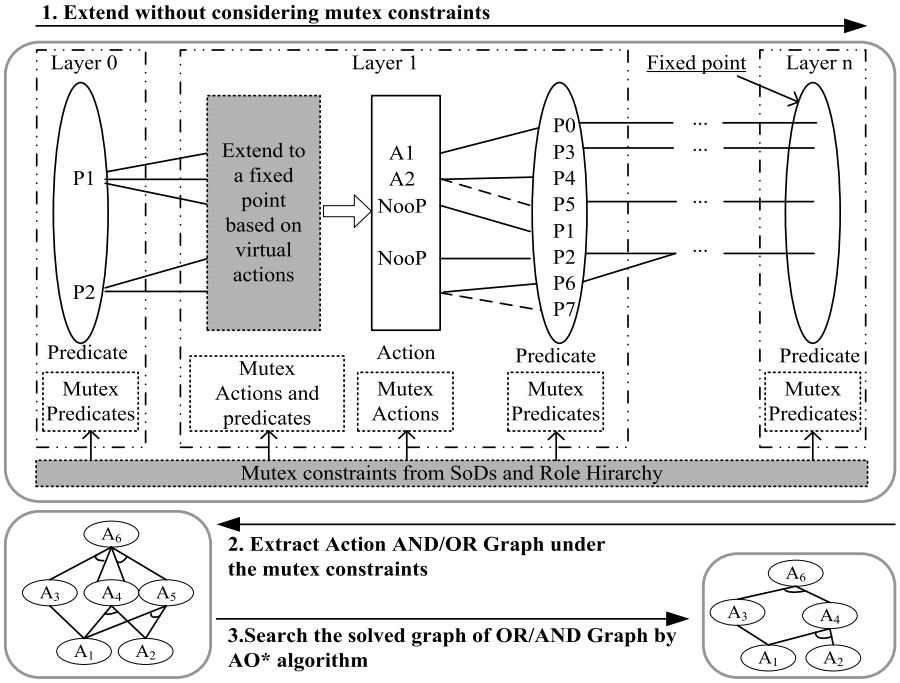



**FIGURE 2.** The framework of PGAO* algorithm.

Another relevant topic is the policy formalization. Our work is based on three policy components of ARBAC97. These components provide a formalization approach to express both assigning of authorization rights and revoking rules. However, there are other formalization approaches for security policy. Joshi et al. presented a novel formalized access control language [25], which was used by Shafiq et al. to design a series of policy integration methodologies in order to perform the policy comparison, merging and restructuring [26]. Sun et al. extended the description capability of RBAC policies and designed algorithms for business logic that formulate security policies [27]. Mentioned formalization approaches attempt to extend the expressive capability of policies in face-specific applications. The ARBAC policy components belong to administrative policy that is used to specify the management rights of administrators. Nonetheless, there is a little difference in formalization language and expressive framework between these approaches.

## III. PROBLEM DEFINITION

The access control in the collaborative work environment is presented in Fig. 1, wherein domains $A$, $B$, and $C$ share their resources on MIoT collaborative platform.

For instance, the dashed lines in the upper part show that users of Factory (A) need to browse or operate the resources of Design Ent. (B), while users of Design Ent. (B) need to browse or operate the resources of Checking Ent. (C). However, the domains have independent authorization strategies that need to be followed respectively. Therefore, a new access control model is constructed, as shown in the lower part of Fig. 1, wherein User Set, Role Set, Permission Set and Policy Set of the model come from three different domains.

Now, we focus on access control and authorization optimization of access control model.

The following definitions are pertinent for AROP problem based on the state space theory and intelligent planning theory.

*Definition 1: Authorization State of RBAC Model:*

An authorization state of RBAC model can be expressed as $S = \{s | s \in 2^{UA \cup PA \cup RH}$, where $UA$ is the assignment relation set that associates users with roles, $PA$ is the assignment relation set that associates permissions with roles, and $RH$ is the inherited relation set among roles [1]–[2].

*Definition 2: Authorization Action:*

An authorization action is a tuple $\langle name, precond, effect \rangle$, where *name* denotes the assigning action, *precond* is the precondition for authorization action (expressed in the Conjunctive Normal Form (CNF)), and *effect* is the performed effect of the authorization action.

*Definition 3: Authorization Route of RBAC Model:* For current state $s_0$, goal role $r_i \in R$, and goal user $u_i \in U$, if there is a legal action sequence $\pi = \{a_1, a_2, \ldots, a_n$ that can transfer $s_0$ to $s_g$, where $s_g = (u_i, r_i)$, then $\pi$ represents an authorization route from $s_0$ to $s_g$. Here, 'legal' means that the execution of the authorization action sequences does not violate SoD rules of RBAC model. More generally, for goal permission $p_i \in P$ and user $u_i \in U$, if there is a legal action sequence $\pi'$ that can transfer $s_0$ to $s_g$, where $s_g' = (p_i, r_i)$

***solve***$(A, IR, GR, D, C)$

1) Define a new ArrayList $G$ to save the planning graph, and add $IR$ to $G$ as an initial layer.
2) ***expandPlanningGraph***$(G, A, D, C, 0)$
3) Create a virtual root node $GNode$.
4) Add $GR$ to $GNode$ as an AND child node.
5) Create a new ArrayList $currentNodeList$ and add the elements of $GR$ to it.
6) ***extractANDORGraph***$(G, currentNodeList, G.size)$
7) ***AOStar***$(GNode)$
8) Output the solved AND/OR graph based on these pointscreated during AO * search process.

*end*

**FIGURE 3.** The *solve* function.

and $s_0, s_1, \ldots \in 2^{UA \cup PA \cup RH}$, then $\pi'$ also represents an authorization route from $s_0$ to $s_g'$.

*Definition 4: Authorization Route Planning Problem (ARPP) of RBAC Model:*

For authorization action set SoD, current state $s_0$, goal role $r_i \in R$, and goal user $u_i \in U$ or a goal permission $p_i \in P$, an authorization route $\pi$ that can transfer $s_0$ to $s_g$, where $s_g = (u_i, r_i)$ or $s_g = (p_i, r_i)$ should be determined.

*Definition 5: Authorization Route Optimization Problem of RBAC Model:*

For authorization action set SoD, current state $s_0$, goal role $r_i \in R$, and goal user $u_i \in U$ or a goal permission $p_i \in P$, the best authorization route $\pi_{best}$ that can transfer $s_0$ to $s_g$, where $s_g = (u_i, r_i)$ or $s_g = (p_i, r_i)$, should be determined.

According to Def. 5, the best authorization route depends greatly on cost definition and calculation mode of authorization route, which will be introduced in detail in Section 5. The AROP is a classical optimization problem.

Based on aforementioned definitions, we can design AROP model by a series of pre-execution steps.

*Step 1:* Instantiate the relevant policies in authorization action set.

In this process, the relationships between authorization actions and their original policies are rerecorded in order to determine the executor of authorization action.

*Step 2:* Express the role hierarchy relationship with virtual action set.

A virtual action is an action that is executed automatically when its precondition is satisfied. In this step, a virtual action is used to express the inherit relationship between two roles.

*Step 3:* Define the initial state and goal state.

The initial state is defined according to current roles played by user, and the goal state is defined according to current user and goal role.

***expandPlanningGraph***$(G, A, D, C, i)$

01) $A_D \leftarrow \{a \in D | canbeAppliedAction(G, a)\}$
02) if not $fixedPoint(G)$then
03) $\quad G \leftarrow expandPlanningGraph(G, A_D, \emptyset, C, i+1)$
04) if $A$ is $\emptyset$ thenreturn $G$
05) $A_i \leftarrow \{a | a \in A \wedge canbeAppliedAction(G, a)\}$
06) $P_i \leftarrow \{p | \exists a \in A_i : p \in effects^+(a)\}$
07) $\mu A_i \leftarrow \{(a, b) \in A_i^2, a \neq$
$\quad b | effects^-(a) \wedge [precond(b) \vee effects^+(b)] \neq \emptyset$
$\quad or\ effects^-(b) \wedge [precond(a) \vee effects^+(a)] \neq \emptyset$
$\quad or\ \exists (p, q) \in G.C_{i-1} : p \in precond(a), q \in precond(b)\}$
08) $\mu P_i \leftarrow \{(p, q) \in P_i^2, p \neq q | \forall a, b \in A, a \neq b:$
$\quad p \in effects^+(a), q \in effects^+(b) \Longrightarrow (a, b) \in \mu A_i$
09) $\mu A_i \leftarrow \mu A_i \cup getRelavantMutex(C, P_i)$
10) for each $a \in A_i$ do: lin a with precondition arcs to $precond(a)$ in $G.P_{i-1}$ and positive arcs to $effects^+(a)$ in $G.P_i$
11) $G \leftarrow G \cup \langle A_i, \mu A_i, P_i, \mu P_i \rangle$
12) if $isVirtualAction(a)$ then $A \leftarrow A - \{a\}$
13) if not $fixedPoint(G)$ then
14) $\quad G \leftarrow expandPlanningGraph(G, A, D, C, i+1)$
15) return $G$

*end*

**FIGURE 4.** The *expandPlanningGraph* function.

*Step 4:* Extend SoDs based on role hierarchy relationship.

It is essential to extend SoD relationships based on role hierarchy relationship. For instance, if $\langle p, q \rangle$ is a mutually exclusive relationship, where $p$ and $q$ are roles, and role $r$ inherits another role from $q$, then $\langle p, r \rangle$ is also a mutually exclusive relationship. These extended SoD relationships are regarded as domain constraints and considered in solving of AROP.

## IV. SOLUTION ALGORITHM

In this section, a new algorithm named PGAO*, which solves AROP by integration of Planning-Graph method [14], [28], and AO* algorithm are presented. In PGAO* algorithm, the Planning-Graph technique is employed to extend RBAC states to the fixed point, the farthest reachable RBAC state, without consideration of constraints from SoDs. Here, the layers between initial layer and fixed-point layer are determined form the extended planning graph of RBAC states. Each layer of planning graph includes two sub layers, the predicate layer and the action layer. In respect to the planning graph of RBAC states, we call these two sub-layers the roles section and the authorization action section, respectively. Each roles section acts as a precondition set of action layers

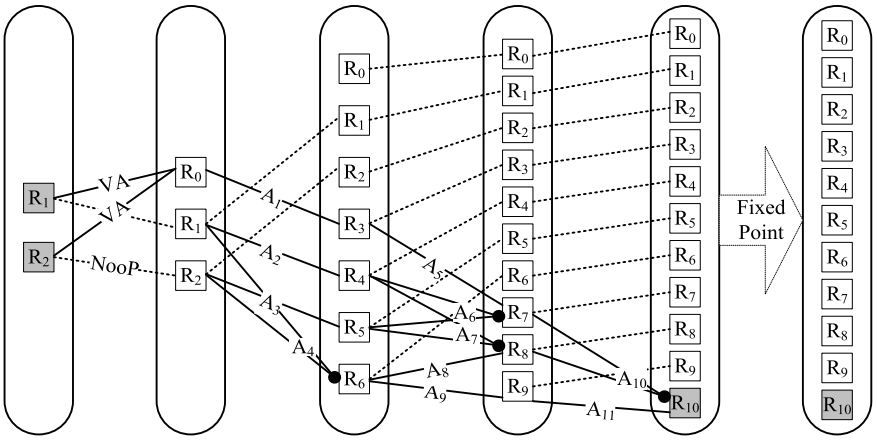



**FIGURE 5.** The planning graph for *expandPlanningGraph* function.

***extractANDORGraph***$(G, CurrentNodeList, no)$

1) if $no < 0$ then return
2) for each $node \in CurrentNodeList$ do:
3) for each $p \in node.precond$ do:
4) for each $a \in G.\ A_{no-1}$ do:
5) if $p \in a.effect$ then
6) $preNodeList \leftarrow preNodeList + \{a\}$
7) $newCurrentNodeList \leftarrow newCurrentNodeList + \{a\}$
8) end
9) $subNodeList \leftarrow subNodeList + \{preNodeList\}$
10) end
11) calculate the Cartesian product of the element set of $subNodeList$
12) judge if each element of the Cartesian product set conflicts with these constraits of the current action section of $G$. If not, regard the element as AND/OR child node of $node$
13) end
14) $extractANDORGraph(G, newCurrentNodeList, no - 1)$

*end*

**FIGURE 6.** The *extractANDORGraph* funtion.

of the next layer. Further, each authorization action, if its precondition is implied by current roles section, can be added to the authorization action of the next layer, and its effects can be added to the roles section of the next layer. The corresponding relationships between authorization action section and roles section are maintained completely.

The framework of PGAO* algorithm is presented in Fig. 2. When the planning graph is extended, it can be noticed that before any extension performed by actions, there is always an extra extension performed in advance by virtual actions in order to ensure that the role hierarchy relationship can be used at a proper time. Once the extending process is complete, we can determine if the goal state is included in the fixed point. If it is not included, then it fails to solve the AROP; otherwise, an extracting process is executed to create an AND/OR graph.

During this process, constraints from SoDs are considered and authorization actions are regarded as nodes, while AND/OR relationships among adjacent nodes are extracted via the extended planning graph. Finally, a standard AO* algorithm is employed to find an optimal solution for AND/OR graph.

In order to explain the solving process of AROP using the proposed PGAO∗ algorithm, an example is provided in the

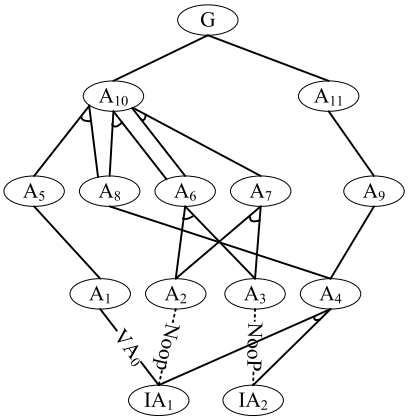



**FIGURE 7.** The AND/OR graph generated by *extractANDORGraph* funtion.

following.

**Role hierarchy**: $r_1 > r_0, r_2 > r_0$; //Virtual Action VA

**SODs:** $< r_3, r_5 >$

**Policy set**:

- *can_assign*$(AR, r_0, r_3)$; // generates action A1
- *can_assign*$(AR, r_1, r_4)$; // generates action A2
- *can_assign*$(AR, r_2, r_5)$; // generates action A3
- *can_assign*$(AR, r_1 \bigwedge r_2, r_6)$; // generates action A4
- *can_assign*$(AR, r_3, r_7)$; // generates action A5
- *can_assign*$(AR, r_4 \bigwedge r_5, r_7 \bigvee r_8)$; // generates actions A6 and A7
- *can_assign*$(AR, r_7 \bigwedge r_8, r_{10})$; // generates action A10
- *can_assign*$(AR, r_6, r_8)$; // generates action A8
- *can_assign*$(AR, r_6, r_9)$; // generates action A9
- *can_assign*$(AR, r_9, r_{10})$; // generates action A11

**Initial roles:** $r_1, r_2$

**Goal role:** $r_{10}$

The presented example shows how the user, who is *Designer* $(r_1)$ and *Process Engineer* $(r_2)$ gets the role *General Manager* $(r_{10})$; please see Fig. 1.

In the following, several main functions of PGAO* algorithm are explained:

*solve* $(A, IR, GR, D, C)$ is the main solving function, where $A$ is the authorization action set, $IR$ is the initial role set assigned to the current user, $GR$ is the goal role assigned to the current user, $D$ is the virtual action set, and $C$ is the constrain set, Fig. 3.

The function *expandPlanningGraph*$(G, A, D, C, i)$, where in the parameter $G$ is the planning graph and $i$ is the operational layer number in the planning graph, expands the planning graph using the iteration process, Fig. 4.

In Fig. 4, the lines from 1 to 4 extend the planning graph with virtual actions; lines from 5 to 12 extend the planning graph with actions; line 7 updates the mutex actions and line 8 updates mutex predicates; line 9 handles the domain constraints from SODs and adds them to mutex action set of every layer of the planning graph via a pre-proceed process named *getRalvantMutex;* lines 10 and 11 configure the planning graph and solution; line 12 trims the performed virtual actions; and lastly, lines 13 and 14 perform the expanding process recursively.

When *expandPlanningGraph* function is applied to the aforementioned case, the output is a planning graph presented in Fig. 5. In Fig. 5, dotted lines are used to express the empty operations labeled as 'NooP', which means that no action is performed. The function *extractANDORGraph*$(G, CurrentNodeList, no)$ is used to extract the AND/OR graph based on the planning graph $G$ using iteration process in which each action is regarded as a node, Fig. 6. In this function, *CurrentNodeList* represents the AND/OR node set that await extension, and *no* is the current operational layer number in the planning graph, which differs from parameter $i$, namely, the former is decreasing and the latter is increasing.

In Fig. 6, the lines from line 2 to line 10 are used to extract all possible precondition-satisfied relationships between current action and actions of previous layer in the expanded planning graph; and line 11 and line 12 are used to refine the AND/OR relationship between current action and previous actions using the Cartesian product calculation and constraint-satisfied calculus. It should be highlighted that this refining process is still an iteration process even though we described this process verbally in lines 11 and 12. The Cartesian product calculation is performed on element sets in *subNodeList* since each element of *subNodeList* is still a set. We first calculate the Cartesian product of the first two element sets in *subNodeList*, and then, we calculate the Cartesian product of created Cartesian product set and the next element set in *subNodeList*, and perform this calculation iteratively. The combinatorial explosion has to be considered in each iterative process, so the constraints of current action section are used to reduce the size of Cartesian product set during each iteration.

Based on generated planning graph, the function *extractANDORGraph* can extract the AND/OR graph as shown in Fig. 7.

*AOStar*$(GNode)$is a function for finding of the optimal solution graph based on AND/OR graph, where *GNode* is the root node of AND/OR graph, and its child nodes are saved in a list as members of *GNode*. In general algorithm, the pseudo code of AO* is ignored.

The result generated by AO* depends on relationship between each node cost and authorization action cost. The method for calculation of authorization action cost depends on optimization criterion defined before solving of AROP. Thus, if we define different optimization criteria, PGAO* algorithm will provide different solutions.

## V. EVALUATION OF AUTHORIZATION ROUTES

Since one or more roles should to be assigned to goal user in the authorization route, the permissions are uniformly distributed to goal role or goal user. Here, we attempt to reduce the spread of extra permissions by authorization route

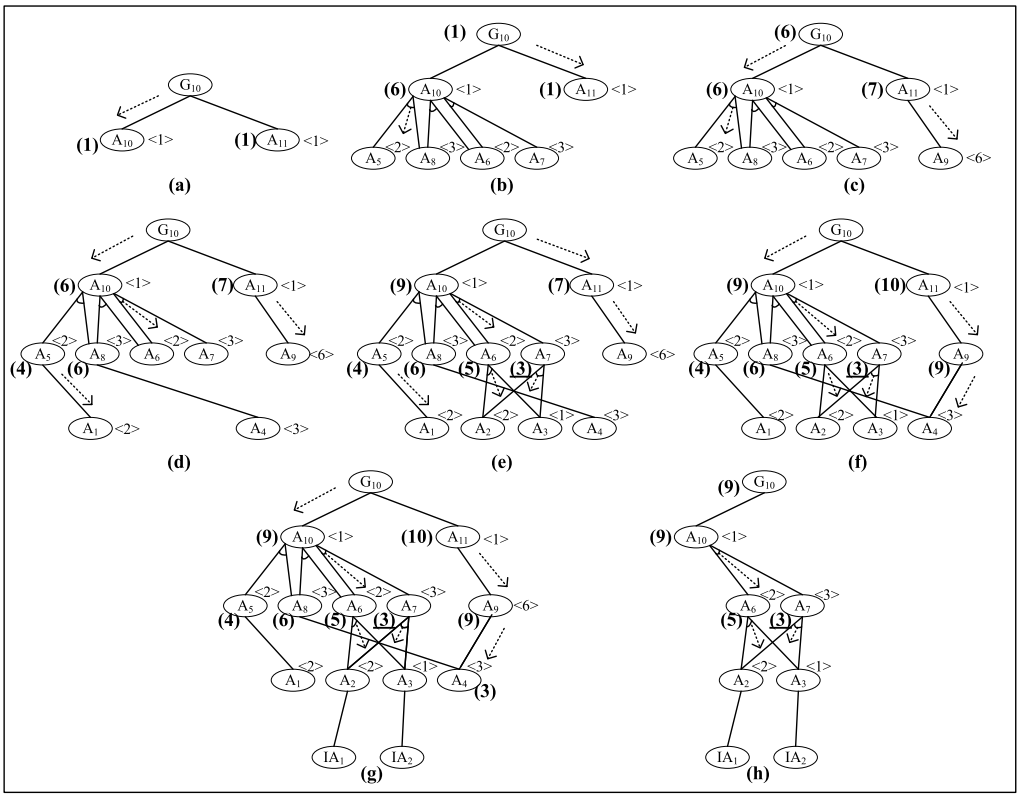



**FIGURE 8.** The search process of optimal authorization route in AND/OR Graph.

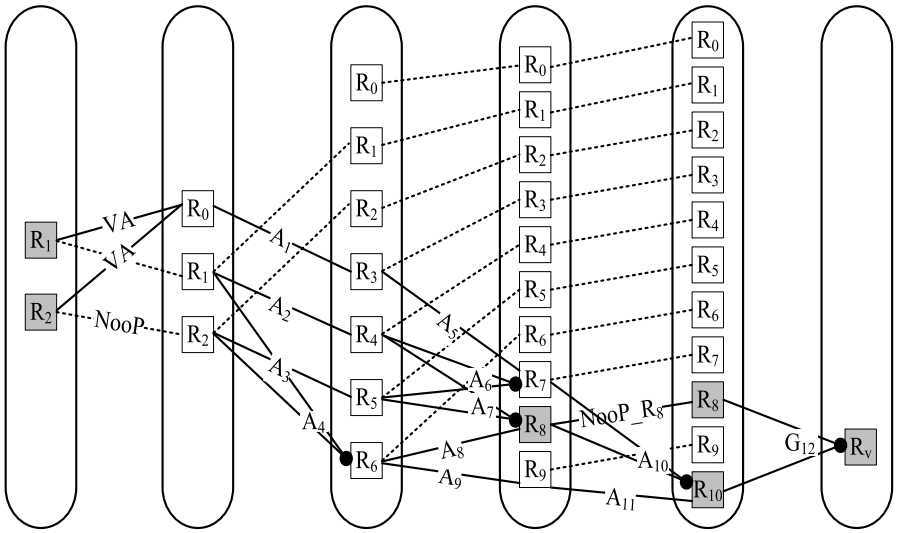



**FIGURE 9.** The planning graph of the multi-goal-role AROP.

optimization. In order to find the optimal route, a quantitative evaluation method is designed to evaluate the quality of authorization route. In this method, the basic unit of measurement is the number of permissions, and each authorization action is assigned to the cost, whose value refers to the effect of number of permissions on the role.

Now, we will explain the authorization action cost calculation on the example presented in Section 4. If we assume that: role $r_0$ has one permission, $r_1$ has two permissions, $r_2$ has two permissions, $r_3$ has two permissions, $r_4$ has two permissions, $r_5$ has one permission, $r_6$ has three permissions, $r_7$ has two permissions, $r_8$ has three permissions, $r_9$ has

six permissions, $r_{10}$ has one permission, and every role has different permissions; then, as it is shown in Fig. 5, the cost of action 1 is 2 since it effects $r_3$. Consequently, the cost of action 2 is 2, the cost of action 3 is 1, the cost of action 4 is 3, the cost of action 5 is 2, the cost of action 6 is 2, the cost of action 7 is 2, the cost of action 3 is 2, the cost of action 8 is 3, the cost of action 9 is 6, the cost of action 10 is 1, and lastly, the cost of action 11 is 1.

Based on AND/OR graph presented in Fig. 7, $AO^*$ function is used to determine the optimal authorization route. The route determination process is shown in Fig. 8, and the optimal authorization route is shown in Fig. 8(h), wherein the user who is a member of $r_1$ and $r_2$ can be assigned to $r_{10}$ with the lowest cost of 9. As it can be seen in Fig. 8, the authorization route begins with the role set $\{r_1, r_2\}$, then goes through $\{r_4, r_5\}$ and $\{r_7, r_8\}$, and ends at $\{r_{10}\}$. According to Fig. 8 and Fig. 1, the user who is *Designer* ($r_1$) and *Process Engineer* ($r_2$) can get the role *General Manager* ($r_{10}$) by applying of roles *Design Supervisor* ($r_4$) and *Process Supervisor* ($r_5$) firstly, and then by applying of roles *Vice President of Quality Assurance* ($r_7$) and *Vice President of Product* ($r_8$) when the roles *Design Supervisor* ($r_4$) and *Process Supervisor* ($r_5$) are authorized.

Nonetheless, in authorization route, the roles might have common permissions and some permissions might be included by other permissions. For instance, file reading permission is often included in file updating permission; thus, it might be complicate to calculate the exact authorization route cost. Therefore, checking of duplicated permissions in authorization route should be obtained before the cost is calculated.

More generally, some permissions might be more important than others. Hence, a weighted cost calculation mechanism based on number of permission is necessary. The administrator can define the permission weighting by number or grade. However, even in the weighted cost calculation mechanisms, checking of duplicated permissions is still necessary.

## VI. MULTI-GOAL-ROLE AROP

The additional advantage of PGAO* algorithm refers to support for multiple-goal-role AROPs. During the solving of such AROP, extra virtual node that links goal nodes after expanding of AROP planning graph, should be added. For instance, in aforementioned example in Section 4, if extra role $r_8$ is also a goal role, then AROP is a multi-goal-role AROP.

The extended planning graph of multi-goal-role AROP is shown in Fig. 9, wherein in contrast to the planning graph presented in Fig. 5, a virtual node $r_v$ is added and virtual authorization action $A_{12}$ is constructed to link two goal roles to final and virtual goals. Therefore, the multi-goal-role AROP is modeled as a single-goal-role AROP, thus, the extracting and searching processes need no change. The final AND/OR graph and optimal authorization route of the multi-goal-role AROP are shown in Fig. 10. The optimal authorization route has a cost value of 13. Here, the user who is *Designer* ($r_1$) and *Process Engineer* ($r_2$) can get the roles *General Manager* ($r_{10}$) and *Vice President of Product* ($r_8$) by applying of role *President of Technology Department* ($r_6$) firstly, and then by applying of roles *Vice President of Product* ($r_8$) and *Vice President of Technology* ($r_9$) when the role *President of Technology Department* ($r_6$) is authorized.

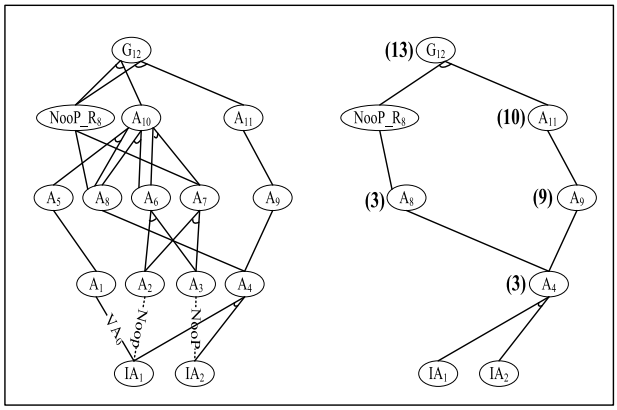



**FIGURE 10.** The AND/OR graph and solution of the multi-goal-role AROP.

## VII. AROP COMPUTATIONAL COMPLEXITY

In the modeling process presented in Section 4, ARPP can be regarded as a classical planning problem as well as a NPC problem [28]. AROP requires the best authorization route, which fundamentally represents an optimization problem with the same problem construction as ARPP. According to [29], the planning problem has the same computational complexity as the isomorphic optimization problem, thus, AROP is also a NPC problem.

In this subsection, two simple policies for engineering applications are presented.

The first policy is based on the following steps. The first step is to define the reasonable cardinality constraint of RBAC. Then, if the number of roles that can be assigned to certain user is limited, the search process is trimmed and the search space is contracted. For instance, if a cardinality constraint of RBAC is set to restrict the user to just three roles, then the search of AND/OR graph of authorization action is trimmed, Fig. 11. In Fig. 11(d), it is shown that any extension from action 5 to action 8 is in conflict with given cardinality constraint, thus, the following searches are interrupted. We used the symbol '=' to represent a trimming operation. We also instituted a trimming propagation process, which specifies that node is trimmed if all of its child nodes are trimmed. Accordingly, action 10 can be trimmed. At the end, we can get an optimal authorization route without disobeying the cardinality constraint, as shown in Fig. 11(g).

The second policy is as follows. The first step is to conduct the correlation analysis between given AROP problem and corresponding policies, roles and permissions. The second step is to select policies, roles, and permissions relevant only to AROP problem. Namely, each policy has its action spheres, such as access control domain. However, all policies are

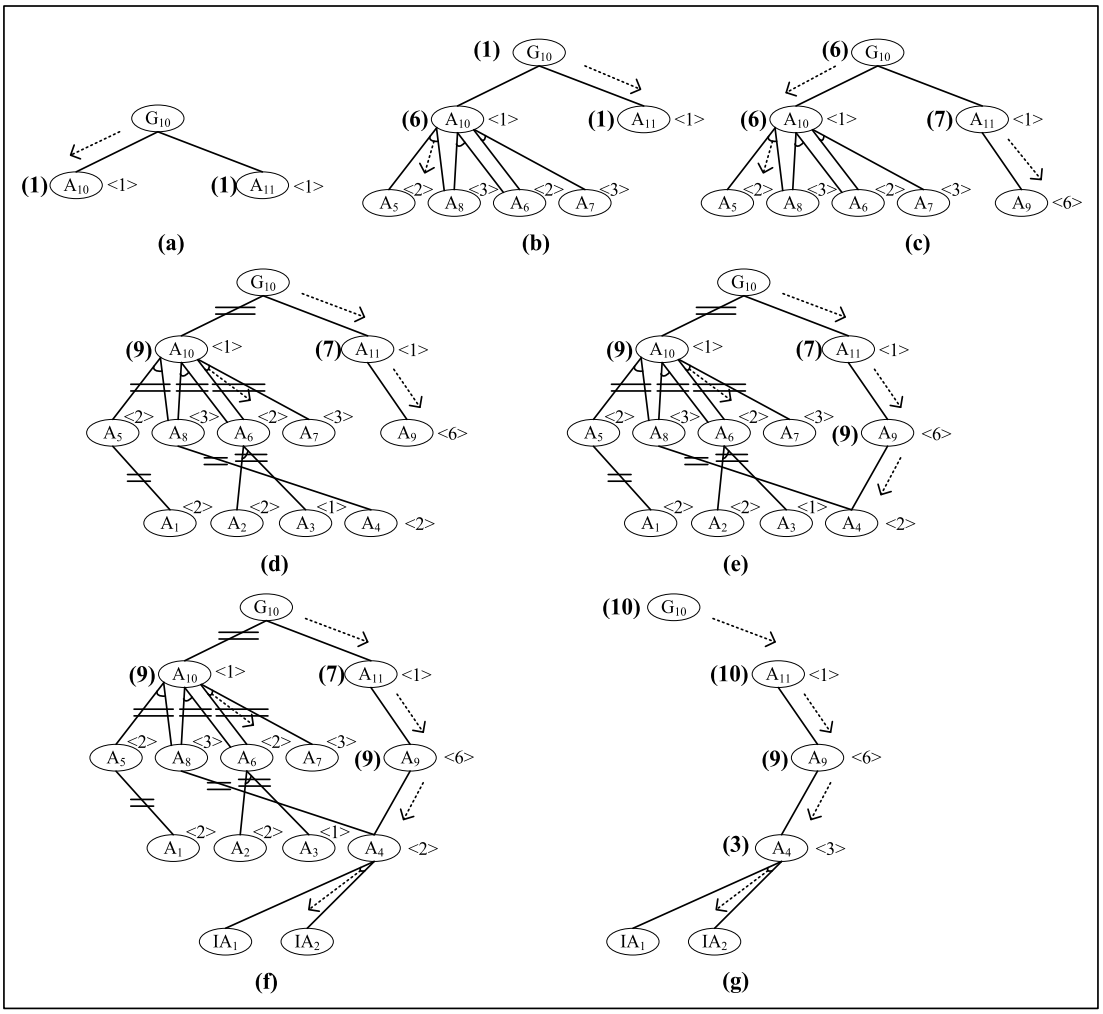



**FIGURE 11.** The searching process of AND/OR graph with trimming operations.

not valid to the specific AROP. Thus, it is not necessary to initialize all security policies, roles and permissions. If the numbers of policies, roles and permissions are limited, the problem space of the specific AROP shrinks concordantly because the number of layers and the scale of each layer decrease when AROP planning graph is extended.

## VIII. CONCLUSIONS

In this paper, an access control model for resource sharing based on the Role-Based Access Control intended for multi-domain MIoT is proposed. In addition, AROP and PGAO* algorithms are designed. The proposed model and algorithms can help administrators to make an accurate decision, decrease the workloads, and strengthen the access safety in resource sharing.

Although, the proposed model provides the optimal authorization routes for received authorization requests, in certain cases it does not have a perfect performance, thus, more factors need to be considered in order to improve the model. The best solution would be an authorization decision support system with a powerful interactive capability that helps administrators to make decision, and an automatic authorization machine that helps administrators to perform the authorization actions.

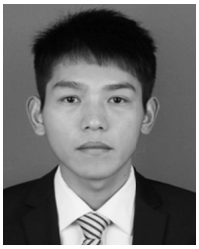
Future research will be focused on automatic authorization mechanisms [30] for collaborative multi-domain RBAC model. However, in making of authorization decision, it is impossible and unreasonable to replace administrator completely with an agent or intelligent program.

## REFERENCES

[1] D. Zhang, Z. He, Y. Qian, J. Wan, D. Li, and S. Zhao, "Revisiting unknown RFID tag identification in large-scale Internet of Things," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 24–29, May 2016.
[2] T. Qiu, X. Liu, L. Feng, Y. Zhou, and K. Zheng, "An efficient tree-based self-organizing protocol for Internet of Things," *IEEE Access*, vol. 4, no. 6, pp. 3535–3546, 2016.
[3] T. Qiu, N. Chen, K. Li, D. Qiao, and Z. Fu, "Heterogeneous Ad Hoc networks: Architectures, advances and challenges," *Ad Hoc Netw.*, vol. 55, pp. 143–152, Feb. 2017.
[4] T. Qiu, D. Luo, F. Xia, N. Deonauth, W. Si, and A. Tolb, "A greedy model with small world for improving the robustness of heterogeneous Internet of Things," *Comput. Netw.*, vol. 101, pp. 127–143, Jun. 2016.
[5] J. Wan, H. Yan, D. Li, K. Zhou, and L. Zeng, "Cyber-physical systems for optimal energy management scheme of autonomous electric vehicle," *Comput. J.*, vol. 56, no. 8, pp. 947–956, 2013.

[6] J. Wan *et al.*, "Software-defined industrial Internet of Things in the context of industry 4.0," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7373–7380, Oct. 2016.
[7] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data based feedback and coordination," *Comput. Netw.*, vol. 101, pp. 158–168, Jun. 2016.
[8] A. J. Jara, M. A. Zamora-Izquierdo, and A. F. Skarmeta, "Interconnection framework for mHealth and remote monitoring based on the Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 47–65, Sep. 2013.
[9] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: Perspectives and challenges," *Wireless Netw.*, vol. 20, no. 8, pp. 2481–2501, Nov. 2014.
[10] A. Gluhak, S. Krco, and M. Nati, "A survey on facilities for experimental Internet of Things," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 58–67, Nov. 2011.
[11] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Comput.*, vol. 29, no. 2, pp. 38–47, Feb. 1996.
[12] R. Sandhu, R. Bhamidipati, and R. Munawer, "The ARBAC97 model for role-based administration of roles," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 1, pp. 105–135, 1999.
[13] S. Oh and R. Sandhu, "A model for role administration using organization structure," in *Proc. ACM Symp. Access Control Models Technol.*, 2002, pp. 155–162.
[14] S. Oh, C. W. Byun, and S. Park, "An organizational structure-based administration model for decentralized access control," *J. Inf. Sci. Eng.*, vol. 22, no. 6, pp. 1465–1483, 2006.
[15] M. Harrison, W. Ruzzo, and J. Ullman, "Protection in operating systems," *Commun. ACM*, vol. 19, no. 8, pp. 461–471, 1976.
[16] N. Li, J. Mitchell, and W. Winsborough, "Beyond proof-of-compliance: Safety and availability analysis in trust management," *J. ACM*, vol. 52, no. 3, pp. 474–514, 2005.
[17] N. Li and M. V. Tripunitara, "Security analysis in role-based access control," *ACM Trans. Inf. Syst. Secur. (TISSEC)*, vol. 9, no. 4, pp. 391–420, 2006.
[18] S. Jha, N. Li, M. Tripunitara, Q. Wang, and W. Winsborough, "Towards formal verification of role-based access control policies," *IEEE Trans. Depend. Sec. Comput.*, vol. 5, no. 4, pp. 242–255, Oct. 2008.
[19] A. Sasturkar *et al.*, "Policy analysis for administrative role based access control," in *Proc. IEEE Workshop Comput. Secur. Found.*, Venice, Italy, Jul. 2006, pp. 183–196.
[20] S. D. Stoller, P. Yang, C. Ramakrishnan, and M. I. Gofman, "Efficient policy analysis for administrative role based access control," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Alexandria, Egypt, 2007, pp. 445–455.
[21] L. Cholvy and F. Cuppens, "Analyzing consistency of security policies," in *Proc. IEEE Symp. Secur. Privacy*, Oakland, CA, USA, May 1997, pp. 103–112.
[22] H. Frode and O. Vladimir, "Conformance checking of RBAC policy and its implementation," *Lect. Notes Comput. Sci.*, vol. 34, no. 39, pp. 144–155, 2005.
[23] F. Huang, Z. Huang, and L. Liu, "A DL-based method for access control policy conflict detecting," in *Proc. 1st Asia–Pacific Symp. Internetware*, Beijing, China, 2009, pp. 1–5.
[24] J. Crampton and H. Khambhammett, "A framework for enforcing constrained RBAC policies," in *Proc. Int. Conf. Comput. Sci. Eng.*, Vancouver, BC, Canada, 2009, pp. 195–200.
[25] J. B. D. Joshi, "Access-control language for multidomain environments," *IEEE Internet Comput.*, vol. 8, no. 6, pp. 40–50, Dec. 2004.
[26] B. Shafiq, J. B. D. Joshi, E. Bertino, and A. Ghafoor, "Secure interoperation in a multidomain environment employing RBAC policies," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1557–1577, Nov. 2005.
[27] Y. Q. Sun, X. Meng, B. Gong, Z. Lin, and E. Bertino, "Specification and enforcement of flexible security policy for active cooperation," *Inf. Sci.*, vol. 179, no. 15, pp. 2629–2642, 2009.
[28] A. Blum and M. L. Furst, "Fast planning through planning graph analysis," *Artif. Intell.*, vol. 90, nos. 1–2, pp. 281–300, 1995.
[29] T. Bylander, "Complexity results for planning," in *Proc. Int. Joint Conf. Artif. Intell.*, San Francisco, CA, USA, 1991, pp. 274–279.
[30] R. Sandhu and V. Bhamidipati, "The ASCAA principles for next-generation role-based access control," in *Proc. 3rd Int. Conf. Availability, Rel. Secur.* Barcelona, Spain, 2008, pp. 27–32.

**QIANG LIU** received the B.S. degree from Xi'an Ploytechnic University, the M.S. degree from the Guangdong University of Technology, and the Ph.D. degree from SunYat-Sen University. He is currently a Professor with the Guangdong University of Technology. His research interests include access control and intelligent manufacturing.



**HAO ZHANG** received the B.A. degree in mechanical engineering from the Hunan Institute of Science and Technology, China, in 2013. He is currently pursuing the Ph.D. degree with the Guangdong University of Technology, China. His research interests include access control and intelligent manufacturing.

**JIAFU WAN** (M'11) has been a Professor with the School of Mechanical & Automotive Engineering, South China University of Technology, since 2015. He has directed 12 research projects, including the National Natural Science Foundation of China, the High-level Talent Project of Guangdong Province, and the Natural Science Foundation of Guangdong Province. He has authored or co-authored over 70 journal papers (with 60+ indexed by ISI SCIE) and 30 international conference papers, with a total of over 2500 citations, an h-index of 26 and an i10-index of 47, according to Google Scholar Citations. He has authored or co-authored eight ESI Highly Cited Papers and three ESI Hot Papers according to Web of Science. His research interests include cyber-physical systems, industry 4.0, smart factory, industrial big data, industrial robot, and internet of vehicles. He is a Senior Member of CMES and CCF. He is the General Chair at the 2016 International Conference on Industrial IoT Technologies and Applications (IndustrialIoT 2016) and 7th EAI International Conference on Cloud Computing (CloudComp 2016). His research results have been published in several famous journals, such as the IEEE Communications Surveys and Tutorials, the IEEE Transactions on Industrial Informatics, the IEEE Transactions on Intelligent Transportation Systems, the *IEEE Communications Magazine*, the *IEEE Network*, the IEEE Wireless Communications, the IEEE Systems Journal, the IEEE Sensors Journal, and *ACM Transactions on Embedded Computing Systems*. He is an Associate Editor of the IEEE Access (SCI), and he is a Managing Editor of IJAACS (Ei Compendex) and IJART (Ei Compendex). He is a Guest Editor of several SCI-indexed journals, such as the IEEE Systems Journal, the IEEE Access, *Elsevier Computer Networks*, *Mobile Networks & Applications*, *Computers and Electrical Engineering*, and *Microprocessors and Microsystems*.

**XIN CHEN** received the B.S. degree from Central South University, the M.S. degree from the Harbin Institute of Technology, and the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor with the Guangdong University of Technology. His research interests include electronic equipment manufacturing and coordination and design optimization.