World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:6, 2017

An Attribute Based Access Control Model with POL Module for Dynamically Granting and Revoking Authorizations

Gang Liu, Huimin Song, Can Wang, Runnan Zhang, Lu Fang

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf

***Abstract*—**Currently, resource sharing and system security are critical issues. This paper proposes a POL module composed of $PRIVILEGE$ attribute (PA), obligation and log which improves attribute based access control (ABAC) model in dynamically granting authorizations and revoking authorizations. The following describes the new model termed PABAC in terms of the POL module structure, attribute definitions, policy formulation and authorization architecture, which demonstrate the advantages of it. The POL module addresses the problems which are not predicted before and not described by access control policy. It can be one of the subject attributes or resource attributes according to the practical application, which enhances the flexibility of the model compared with ABAC. A scenario that illustrates how this model is applied to the real world is provided.

***Keywords*—**Access control, attribute based access control, granting authorizations, privilege, revoking authorizations, system security.

I. Introduction

ACCESS control [1] is a technology used in almost every system, such as a computer system or an enterprise management system. It permits legally authorized subjects to access protected resources; prohibits legal subjects from unauthorized access to protected resources; prevents illegal subjects from accessing protected resources. Nowadays, the mainstream access control models are mandatory access control (MAC) [2], discretionary access control (DAC) [3], role based access control (RBAC) [4], task based access control (TBAC) [5] and attributed based access control (ABAC) [6].

Many researchers have investigated ABAC model, which is applied in web service, cloud computing, collaborative environment and so on. Reference [7] utilizes the characteristic that security information is spread over a number of attribute and policy authorities in ABAC model and compares ABAC model with RBAC model to further elaborate on the advantages of ABAC model. Reference [8] takes advantage of the characteristic that policy authorization based attributes can protect the user privacy. In [9], the access control policies are made based the users previous behavior. The much contribution a user makes, the more permission he or she is granted. ABAC is not only applied in different fields but also more flexible than other access control models. ABAC describes the subject, resource, operation and environment with attributes. Each attribute is defined according to the practical application which improves the flexible of ABAC model. ABAC makes the access control policies based one or more attributes and evaluates the value of them to judge whether the subject can access the resource or not. That is to say, the subject can access the resource only through parts of his or her attributes which is different from the other access control models and protects the subject privacy. Besides, the way of authorization based attributes in ABAC model also achieves large-scale authorization and the policies can be modified dynamically.

However, though ABAC model has many merits in application, it is also necessary to take revoking authorization flexibly into consideration. If the authorization cannot be revoked in time, it may carry significant security risks to the system. In the real world, a lot of authorizations are used only once or several times. If these authorizations are stored in the policy repository not revoked for a long time, the problem of the policy repository explosion and policy conflict may occur. Thus, how to revoke the authorizations in time is the research contents in this paper.

This paper proposes a POL module and combines it with ABAC model, termed PABAC model. PABAC is also based on attributes, including subject attribute, resource attribute, operation attribute and environment attribute. The environment attribute has been used to judge the current situation state of the system. POL module is composed of $PRIVILEGE$ attribute (PA), obligation and log. POL module can be one of the subject attributes or resource attributes. This paper takes the POL module as one of the resource attributes as an example to illustrate the structure of PABAC. Due to it, each resource has been managed by the only one subject, termed $PRIVILEGE$ manager (PM). However, a PM manages one or more resources and has the permission to modify the PA value of these resources. When the environment attribute detects that the current system state is abnormal, POL module allows the PM to authorize based one or more subject attributes with modifying the PA. Once the subject has been authorized access, the subject executes the obligation before or after the authorized access. Thus, obligation is used to revoke the $PRIVILEGE$ directly or gives the information feedback of the authorized access. The PA can also be used to address the problems including the unforeseeable problem and the problem not described by access control policies. That is to say, PABAC makes use of the PA to authorize access and the obligation to revoke it which achieves granting and

Huimin Song is with the School of Computer Science and Technology, XIDIAN University, Xi'an, 710071 China (corresponding author, e-mail: $gliu_xd@163.com$).
Gang Liu, Can Wang, Runnan Zhang, Lu Fang are with XIDIAN University.

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf

revoking authorization dynamically. Besides, the authorization to a subject in ABAC needs to make a complex access control policy rule involved a lot of attributes but PABAC don't need. The following illustrates PABAC model in terms of the structure, attribute definition and access control policy in detail.

The paper is organized as follows: Section II interprets the related work including the introduction of ABAC model and obligation; Section III details the core model components, POL module, the policies of access control and obligation in PABAC model; Section VI describes a scenario for illustrating the application method; Section V summarizes the whole thesis and provides the future development tendency.

II. Related Work

A. ABAC

ABAC is a logical access control model that controls access to resources by evaluating rules against the attributes of entities (subject and resource), operations, and the environment relevant to a request [10]. ABAC uses attributes as the basis for authorizations instead of directly defining permissions between subjects and resources. When adding a role as a subject attribute, ABAC is a superset of RBAC [11].

The relations of the attributes and the basic architecture of ABAC in [12] are shown in Fig. 1.

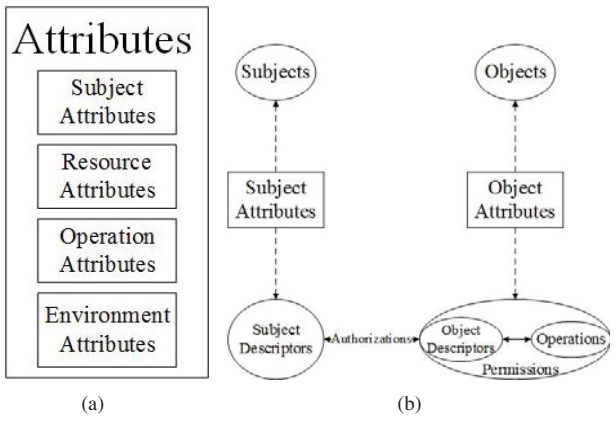


Fig. 1 (a) Relations of attributes. (b) Architecture of ABAC

However, there is also some problems existed in ABAC. When the emergency situation happens, it is difficult to handle with the access control policy predefined. When some authorizations only used once or several times, ABAC cannot revoke them flexibly, which may lead to the problem of the policy repository explosion. Besides, the problem of the policy conflict also exists in ABAC model. PABAC adds a PA in ABAC which can authorize access in abnormal situations and revokes the authorization with obligation. The way to dynamically grant and revoke authorizations addresses the problem of the policy repository explosion. The authorization to a subject in ABAC also needs to make a complex access control policy rule involved a lot of attributes but PABAC don't need.

B. Obligation

Obligations express the actions that must be executed before or after granted access [13]. Thus, obligation has been categorized in two types: provisions are those conditions that need to be satisfied or actions that must be performed before a decision is rendered; obligations are those conditions or actions that must be fulfilled by either the users or the system after the decision [14]. In the following, the provisions have been termed pre-obligation while the obligations have been termed post-obligation.

Obligation can not only control accesses independently but also combine with other access control models. Such as [15] is the combination of obligation and RBAC model. Thus, PABAC introduces the obligation into the ABAC model. When the subject has been granted access, the subject needs to give an information feedback in order to revoke the $PRIVILEGE$ in time. That is to say, when not in use, the $PRIVILEGE$ are revoked by PM and the $PRIVILEGE$ can be assigned to other subjects. Thus, PABAC improves the flexible of revoking authorizations.

III. PABAC Model

This section introduces the PABAC model, in terms of the attribute and policy definition, the POL module in detail. POL module is composed of PA, obligation and log which will be detailed in the following. POL module can be one of the subject attributes or resource attributes which is based practical application. When the number and kind of subjects are enormous, POL module is one of subject attributes; when the number and kind of resources are enormous, POL module is one of resource attributes. These two methods make it easy to add, modify or revoke the $PRIVILEGE$. This paper takes the POL module to be one of the resources attributes as example to introduce the PABAC.

A. Core Model Components

1) Subject Attribute (SA): A subject is a human user or non-person entity, such as a device that issues access requests to perform operations on resources while subjects are assigned one or more attributes [6]. Subject attribute in PABAC is defined as $(s, ...)$. Each subject (abbreviated to s) has a unique id termed subject id in the system. Meanwhile, whether s and other attributes are described as a set of integer or name is determined according to the actual application.

2) Resource Attribute (RA): A resource can be managed by the ABAC system or a requested entity, as well as anything upon which an operation may be performed by a subject including data, applications, services, devices, and networks [6]. Each resource (abbreviated to r) has a unique resource id and owns a POL module. PABAC also needs to know who is the $PRIVILEGE$ administrator of the resource. Thus, the resource attribute in PABAC is defined as $(r, POL, PM, ...)$.

3) Operation Attribute (OA): An operation is the execution of a function at the request of a subject upon a resource [6]. The operation attribute values are represented by a set of unique operation types, termed o in the system. Furthermore, system can define different operation types in different actual

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf

application, such as read, write, edit, delete, copy, execute, modify and so on.

4) Environment Attribute (EA): The definition of the environment attributes in PABAC is operational or situational context in which access requests occur [6]. For environment attributes, attribute authentication is based on conditions in the environment when a request is made in ABAC [16]. Furthermore, the environment attributes are very special attributes according to the detection, including the current time, day of week or the current threat level. In PABAC, EA has been used to detect the current system states including the normal situation and the abnormal situation which is defined in the following. Thus, different access control policies is made according to the different system states. Thus, the environment attribute in PABAC is defined as $(CurrentState(CS), ...)$.

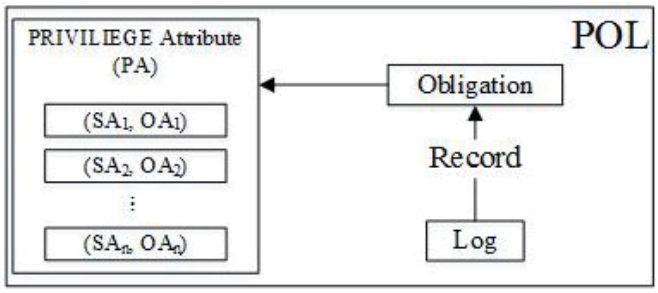


Fig. 2 Architecture of POL module

5) POL Module: Fig. 2 shows that POL module is composed of PA, obligation and log. This paper is taking the POL module to be one of the RA as an example, so the PM who is the owner or manager of the resource can manage its PA. Each resource has only one PM while the PM of the multiple resources can be the same subject.

- $PRIVILEGE$: under the abnormal situation, system authorizes the access based one or more subject attributes though the PA in POL module. PA in POL module is a pair, described as (SA, OA). SA refers to one or more subject attributes while OA refers to one or more operations. If the POL module is one of the subject attributes, the PA is described as (OA, RA). When the subject only want to access the resource once or several times, the PM of the resource authorizes the access through modifying its PA in POL module.
- Obligation expresses the operations that must be executed before or after the authorized access [13]. That is to say, after being authorized access, the subject needs to give an access information feedback to the PM. PM revokes the $PRIVILEGE$ according to the information feedback or obligation revokes the $PRIVILEGE$ directly after a certain time.
- Log records each operation in abnormal situations. When the system administrator finds an unsafe operation in log, the system administrator can take remedial actions to ensure the system security.

The PA in the POL module makes it flexible to manage the permission. Under the abnormal situation, the PM manages the PAs which makes it flexible for subject gaining authorizations. The abnormal situations is common in the real world, so PABAC improves availability of ABAC by introducing the POL module. When some permission used only once or several times, PABAC revokes them by obligations which addresses the problem of the policy repository explosion.

B. Access Control Policy

1) Attribute Expression: Attribute expression is a relation between the attributes and attributes or attributes and attribute values through the mathematical symbols. There are n subject attributes and m resource attributes. $V(SA)$, $V(RA)$ express one of the subject and resource attribute values respectively. The formalized definition of attribute expression is described as follows:

$$\begin{aligned} AE : \{&SA_1 \mid SA_2 \mid ... \mid SA_n \mid RA_1 \mid RA_2 \mid ... \mid RA_m \\ &\mid V(SA_1) \mid V(SA_2) \mid ... \mid V(SA_n) \\ &\mid V(RA_1) \mid V(RA_2) \mid ... \mid V(RA_m)\} \\ &\quad Operator \\ &\{SA_1 \mid SA_2 \mid ... \mid SA_n \mid RA_1 \mid RA_2 \mid ... \mid RA_m \\ &\mid V(SA_1) \mid V(SA_2) \mid ... \mid V(SA_n) \\ &\mid V(RA_1) \mid V(RA_2) \mid ... \mid V(RA_m)\} \end{aligned}$$

$$Operator = \{+, -, =, <, >, \div, \times, \in ...\}$$

For example, the information system defines $User = \{Subjectid = 1,\ Age = 30, ...\}$ and $Resource = \{Id = 4,\ Managerid = 1, ...\}$. The subject id and age of the user refer to SA_1 and SA_2 respectively while 1 and 30 refer to one of the user's id values and age values respectively, so both "$Subjectid(s) = 1$" and "$Age(s) = 30$" are attribute expressions. The manage id of the resource refers to RA_2, so "$Subjectid(s) = Manageid(r)$" is also an attribute expression. Thus, attribute expressions are the way of policy evaluation which determines whether the subject can access or not.

2) PA Set Operation: The PA is designed as a set. PM grants and revokes authorization dynamically through the set operations. There are three resources termed r_1, r_2 and r_3 while $PA(r_1), PA(r_2)$ and $PA(r_3)$ represent the PA sets of r_1, r_2 and r_3 respectively. Thus, the set operations applied among PA sets are defined as follows termed Action:

$$\begin{aligned} Action_1 &: PA(r_1) = PA(r_2). \\ Action_2 &: PA(r_1) = PA(r_2) - PA(r_3) \\ &\quad = \{x \mid x \in PA(r_1) \wedge x \notin PA(r_2)\}. \\ Action_3 &: PA(r_1) = PA(r_2) \cup PA(r_3) \\ &\quad = \{x \mid x \in PA(r_2) \vee x \in PA(r_3)\}. \\ Action_4 &: PA(r_1) = PA(r_2) \cap PA(r_3) \\ &\quad = \{x \mid x \in PA(r_2) \wedge x \in PA(r_3)\}. \end{aligned}$$

$Action_1$ expresses the PA set of r_1 can equal to the PA set of r_2 directly; $Action_2$ expresses the PA set of r_1 can equal to different set between the PA set of r_2 and the PA set of r_3; $Action_3$ expresses the PA set of r_1 can equal to union set between the PA set of r_2 and the PA set of r_3; $Action_4$ expresses the PA set of r_1 can equal to intersection between the PA set of r_2 and the PA set of r_3. The Actions defined above can reduce the operations of defining the same type resource. The PA set also supports to add and delete one PA element. Thus, It is flexible to design PA as a set.

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf

3) Basic Policy Model: The PABAC policy model is defined based on [7] under normal situation (NS):

(1) $SA_k(1 \leq k \leq 4)$, $RA_m(1 \leq m \leq 2)$ and $OA_n(n = 1)$ are the predefined attributes for subjects, resources, and operations, respectively;

(2) ATTR(s), ATTR(r), and ATTR(o) represent attribute assignment relations for subjects, resources and operations, respectively, which address all attribute values:

$$ATTR(s) \subseteq SA_1 \cup SA_2 \cup SA_3 \cup SA_4$$
$$ATTR(r) \subseteq RA_1 \times RA_1$$
$$ATTR(o) \subseteq OA$$

Policy is the representation of rules or relationships that makes it possible to determine if a requested access should be allowed, given the values of the attributes of the subject, resource, and possibly environment conditions. Thus, under the normal situation, the access control policy in PABAC is defined as follows:

$Rule_1$:

$$can_access(s, r, o, e) \leftarrow f(ATTR(s), ATTR(r), ATTR(o), ATTR(e)).$$

4) Abnormal Situations (AS): The abnormal situations includes temporary situation, emergency situation, special situation and so on. Thus, PABAC makes the access control rules based these three situations above as follows:

$Rule_2$:

$$can_access(s, r, o, e) \leftarrow (ID(s) = PM(r) \wedge OA(o) = Action \wedge CS(e) = AS).$$

$Rule_3$:

$$can_access(s, r, o, e) \leftarrow ((s, o) \in PA(r) \wedge CS(e) = AS).$$

When the abnormal situations happen, $Rule_2$ permits the PM modifies the PA values; The value (s, o) can be got by computing "$ATTR(s) \times ATTR(o)$"; $Rule_3$ determines whether the subject has the $PRIVILEGE$ to access the resource through comparing the values of (s, o) and the PA values. That is to say, only when the PM modifies the PA values by $Rule_2$, the subject is granted $PRIVILEGE$ to perform the operation on the resource. Since the subject has been granted the $PRIVILEGE$, the subject must execute the operations defined in obligation before or after the authorized access. Thus, the policy in obligation has been defined according to [13] as follow:

$Rule_4$:

$$Obligation = (Subject, Trigger, Operation).$$

In $Rule_4$, subject refers to who has been granted the authorization by the PA and needs to execute the obligation; trigger refers to the conditions that the subject executes the obligation; operation refers to the operation type that the subject takes. The subject must execute the operations before or after the authorized access, such as when the resource is not used any more, the subject needs to inform the resource PM by signaling or others. The PM deletes the $PRIVILEGE$ in PA values by $Rule_2$. The operations above must be recorded in the log, so the log has been defined as follows:

$Rule_5$:

$$Log = (Subject, Operation, Resource, Action, Time).$$

The time includes the time of the subject doing operation on the resource and executing the obligation.

The authorization in ABAC is based on the evaluation of access control policy rules. This way is difficult to revoke permission that are only used by once or several times. PABAC divides the system into the normal situation and the abnormal situation through the POL module. Under the normal situation, the subject accesses the resource based the access control policies. Under the abnormal situations, the subject obtains the temporal permission through the PM of the resource. The authorization in PABAC is more flexible and dynamic and the permission can be granted and revoked easily by modifying the PA values. The system administrator ensures the system security by checking the log.

C. Architecture of PABAC

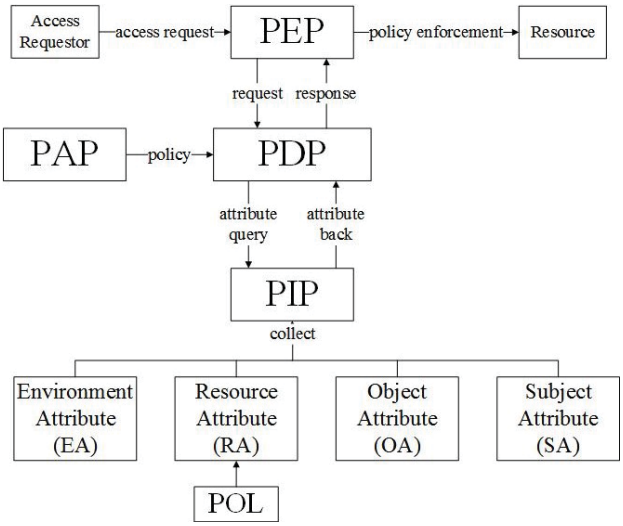


Fig. 3 Architecture of PABAC model

This architecture in Fig. 3 integrates XACML in real implementations. XACML (eXtensible Access Control Markup Language) is an OASIS standard that describes both a policy language implemented in XML and an access control decision request/response language implemented in XML [17]. The XACML profile specifies four main actors in PABAC model to handle access decisions according to [18]:

- Policy administration point (PAP): The system entity creates a policy or policy set.
- Policy information point (PIP): The system entity acts as a source of attribute values. That is to say, PIP is the point where the necessary attributes for the policy evaluation are retrieved from several external or internal actors [19].
- Policy enforcement point (PEP): The system entity performs access control by making decision requests and enforcing authorization decisions.
- Policy decision point (PDP): The system entity evaluates applicable policy and renders an authorization decision.

When the environment attribute detects that the current system state is the normal situation, PIP collects the attributes that is relevant to the access including subject attributes,

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf

resource attributes, operation attributes; PAP looks up the access control policy that is relevant to the access; PDP evaluates the attribute values by access control policy rules and returns the evaluation result to PEP. If the result is true, the access request is permitted; if the result is false, the access request is denied. If there is no access control policy rules that is relevant to the access request, return inapplicable and the access request is denied. When the environment attribute detects that the current system state is the abnormal situation, PIP collects the attributes that is relevant to the access including subject attributes, resource attributes, operation attributes; PDP computes "$SO = SA \times OA$" first and then computes "$SP = SO \cap PA(r)$". If SP is a nonempty set, the access request is permitted and the obligation may be executed; If SP is an empty set, the access request is denied.

IV. An Illustrative Scenario

This section takes an emergency operation in hospital as an example to illustrate PABAC model. The patient is taken to hospital due to falling ill suddenly. The situation cannot be predicted, so the access control policy rules in the normal situation are not appropriate and cannot judge which operating room is unoccupied. The operative time of each operation also cannot be estimated, so the system is hard to manage it with time constraint. This section describes the attributes in scenario first; makes the access control policy rules in the abnormal situation including emergency operation; defines the obligations in this emergency operation; illustrates the mode of operation in the whole scenario.

A. Attribute Definitions

In hospital, all the patients, nurses and doctors have a name, an age and a unique id. Thus, the subject attribute is defined as follows:

$$SA = \{ID, Name, Age\}.$$

ID is composed of a letter and some positive integers. First bit in ID is P, N and D which represents the patient, nurse and doctor respectively; The positive integer assignment is based on when the subject enters into the system. For example, P10 refers to the patient whose number is 10. Name is composed of letters while Age ranges from 0 to 200. All the operating rooms, departments and wards have a unique name and a manager. Thus, the resource attribute is defined as follows:

$$RA = \{Name, Manager, PRIVILEGE\}.$$

Name refers to the resource name or id; Manager refers to the subject ID who manage the resource; $PRIVILEGE$ refers to the PA value set of the resource which is composed of (s, o). Besides, the operation attribute is also defined as follows:

$$OA = \{Type\}.$$

Type includes create, modify, delete and Action which has been described above. The environment attribute in this scenario is also defined as follows:

$$EA = \{CurrentState(CS)\}.$$

The current state values in the system is normal situation (NS) and abnormal situation (AS). The abnormal situation refers to the patient who falls ill suddenly or the situation cannot be predicted before. With a simple analysis, the attributes in the scenario has been defined completely.

B. Access Control Policy

Under the normal situation, hospital manages the subjects and resources with the access control policy rules predefined. The abnormal situation, such as the emergency operation, cannot be predicted before, so hospital needs to manage the subjects and resources flexible. According to the access control policy rules in PABAC, this hospital system defines additional rules as follows:

$Rule_6$:

$can_access(s, r, o, e)$
$\leftarrow (ID(s) = Manager(r)) \wedge Type(o) = Action$
$\wedge\ CS(e) = AS).$

$Rule_7$:

$can_access(s, r, o, e)$
$\leftarrow ((s, o) \in PRIVILEGE(r) \wedge CS(e) = AS).$

$Rule_6$ refers that only when the subject ID is equal to the Manager id of the resource, the subject can take operations which has been defined in the Action above on the value of the resource $PRIVILEGE$. $Rule_7$ computes "$SA \times OA$" to get a set of (s, o) values first, and then judges whether one of the (s, o) values belongs to the $PRIVILEGE$ of the resource to determine whether the subject is authorized. Under the abnormal situation, if one of the (s, o) values belongs to the $PRIVILEGE$ of the resource, the subject is allowed to access the resource; if none of the (s, o) values belong to the $PRIVILEGE$ of the resource, the subject is disallowed to access the resource. Under the normal situation, system evaluates the access control policy rules based attributes. If the evaluation result is true, the subject can access the resource; if the evaluation result is false, the subject cannot access the resource; if there is no access control policy rules that is relevant to the access request, the evaluation result is inapplicable and the subject cannot access the resource.

C. Obligation Policy

This section takes an emergency operation in hospital as an example to define the following obligation policy:

$Policy\ before\ the\ surgery\{$
$Subject = D10.$
$Operation = Occupy.$
$Resource = Operating\ room\ 1.$
$Trigger = Beginning\ of\ operating.$
$Obligation = Turn\ the\ operation\ indicator\ light\ on.$
$\}$
$Policy\ after\ the\ surgery\{$
$Subject = D10.$
$Operation = Occupy.$
$Resource = Operating\ room\ 1.$
$Trigger = Operating\ finished.$
$Obligation = Turn\ the\ operation\ indicator\ light\ off.$
$\}$

The obligation policy above includes pre-obligation and post-obligation. When the doctor whose number is 10 has

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf

been authorized to have an operation in the operating room 1, the doctor must finish the operation defined in the obligation. Doctor 10 gives an information feedback to the manager of the operating room 1 with turning on or off the operation indicator light. The manager modifies the $PRIVILEGE$ of the operating room 1 to revoke the authorization in time according to the information feedback. Thus, obligation makes it easy to manage the authorizations and resources. Compared with ABAC, PABAC revokes authorizations more flexible with obligation. When the permission is not in use any more, the manager of the resource can modify and delete it at once which addresses the problem of the policy repository explosion. PABAC only grants the subject authorizations under the abnormal situation and each operation has been recorded in the log. Thus, PABAC still ensures the system security. Besides, the authorization to a subject in ABAC needs to make a complex access control policy rule involved a lot of attributes but PABAC doesn't need.

V. Conclusion

This paper describes PABAC with granting and revoking authorizations dynamically in detail. PABAC applies the POL module into ABAC model. The manager of the resource grants the subject authorizations immediately with PA in the POL module under the abnormal situation. Besides, obligation in POL module also makes it flexible to revoke the authorization which addresses the problem that the resources in idle state cannot be assigned to other subject. Thus, POL module helps the system to address the problems including not predicted before and not described by access control policy rules. Besides, the authorization to a subject in ABAC needs to make a complex access control policy rule involved a lot of attributes but PABAC doesn't need. Log ensures the system security.

POL module can be one of the subject attributes or one of the resource attributes. When the number and kind of subjects are enormous, POL module is one of subject attributes; when the number and kind of resources are enormous, POL module is one of resource attributes. POL module as different attribute types makes it easy to manage authorization. Thus, PABAC can makes different access control policy rules according to different practical applications which improves the flexible.

Introducing the priority levels into PA is a future research direction. At present, PABAC cannot address the problem of policy conflict in fact. Thus, how to define the priority levels is an urgent problem to be solved, which refines PABAC model.

Acknowledgment

This work is supported by the National Natural Science Foundation (NNSF) of China (Grant No. 61572385).

Conflict of interest statement: there is no conflict of interest regarding the publication of this paper.

References

[1] Garnaut P., Thompson J., "Review of Data Integrity Models in Multi-Level Security Environments," Technical Report DSTO-TN-0971, Defence Science And Technology Organisation Edinburgh Command Control Communications And Intelligence Div, Australia, Feb. 2012.

[2] Alexander P, Pike L, Loscocco P, et al., "Model Checking Distributed Mandatory Access Control Policies," *J. Acm Transactions on Information & System Security,* vol. 18, no. 6, pp. 1-25, Dec. 2015, doi: 10.1145/2785966.

[3] Zamite J, Domingos D, Silva M J, et al., "Group-Based Discretionary Access Control in Health Related Repositories," *J. Journal of Information Technology Research,* vol. 7, no. 1, pp. 78-94, 2014, doi: 10.4018/jitr.2014010106.

[4] Zhou L, Varadharajan V, Hitchens M, "Trust Enhanced Cryptographic Role-Based Access Control for Secure Cloud Data Storage," *J. Information Forensics & Security IEEE Transactions on,* vol. 10, no. 11, pp. 2381-2395, 2015, doi: 10.1109/TIFS.2015.2455952.

[5] Yi Liu, Ke Xu, Junde Song, "A Task-Attribute-Based Workflow Access Control Model," *Proc. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing,* IEEE, pp. 1330-1334, Aug. 2013, doi: 10.1109/GreenCom-iThings-CPSCom.2013.231.

[6] Vincent C. Hu, et al., "Guide to Attribute Based Access Control(abac) Definition and Considerations," National Institute of Standards and Technology, Gaithersburg, 2014.

[7] E. Yuan, J. Tong, "Attributed Based Access Control (ABAC) for Web Services," *Proc. 2005 IEEE International Conference on Web Services(ICWS),* IEEE, pp. 561-569, Jul. 2005, doi: 10.1109/ICWS.2005.25.

[8] Hakima Ould-Slimane, Moustapha Bande, Hanifa Boucheneb, "WiseShare: A Collaborative Environment for Knowledge Sharing Governed by ABAC Policies," *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on,* IEEE, pp. 21-29, Oct. 2012, doi: 10.4108/icst.collaboratecom.2012.250402.

[9] Maryam Ed-Daibouni, Adil Lebbat, Saida Tallal, Hicham Medromi, "Toward a New Extension of the Access Control Model ABAC for Cloud Computing," *Advances in Ubiquitous Networking. Lecture Notes in Electrical Engineering,* Sabir E., Medromi H., Sadik M., eds., Singapore: Springer, pp. 79-89, Feb. 2016, doi: 10.1007/978-981-287-990-5_7.

[10] Vincent C. Hu, D. Richard Kuhn, David F. Ferraiolo, "Attribute-Based Access Control," *J. Computer,* vol. 48, no. 2, pp. 85-88, Feb. 2015, doi: 10.1109/MC.2015.33.

[11] Xu D., Kent M., Thomas L., et al. "Automated Model-Based Testing of Role-Based Access Control Using Predicate/Transition Nets," *J. IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2490-2505, Sep. 2015, doi:10.1109/TC.2014.2375189.

[12] Mike Burmester, Emmanouil Magkos, Vassilis Chrissikopoulos, "T-ABAC: An Attribute-based Access Control Model for Real-time Availability in Highly Dynamic Systems," *Proc. Computers and Communications(ISCC), 2013 IEEE Symposium on,* IEEE, pp. 143-148, Jul. 2013, doi: 10.1109/ISCC.2013.6754936.

[13] Laurent Gomez, Slim Trabelsi, "Obligation Based Access Control," *On the Move to Meaningful Internet Systems: OTM 2014 Workshops. OTM 2014. Lecture Notes in Computer Science,* Meersman R. et al., eds., Berlin: Springer, pp. 79-89, Oct. 2014, doi: 10.1007/978-3-662-45550-0_15.

[14] Claudio Bettini, Sushil Jajodia, X. Sean Wang, Duminda Wijesekera, "Provisions and Obligations in Policy Management and Security Applications," *Proc. VLDB '02 Proceedings of the 28th international conference on Very Large Data Bases,* VLDB Endowment, pp. 502-513, Aug. 2002, doi: 10.1016/B978-155860869-6/50051-2.

[15] Gansen Zhao, David Chadwick, Sassa Otenko, "Obligation for Role Based Access Control," *Proc. Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on,* IEEE, pp. 424-431, May 2007, doi: 10.1109/AINAW.2007.267.

[16] Michael J. Covington, Manoj R. Sastry, "A Contextual Attribute-Based Access Control Model," *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. OTM 2006. Lecture Notes in Computer Science,* Meersman R., Tari Z., Herrero P., eds., Berlin: Springer-Verlag, pp. 1996-2006, Nov. 2006, doi: 10.1007/11915072_108.

[17] Anoop Singhal, Theodore Winograd, Karen Scarfone, "Guide to Secure Web Services," National Institute of Standards and Technology Special Publication, Gaithersburg, 2007.

[18] Bill Parducci, Hal Lockhart, Rich Levinson, "eXtensible Access Control Markup Language (XACML) Version 3.0," Burlington, USA: OASIS, 2013.

[19] Mehdi Sabbari, Hadiseh Seyyed Alipour, "Improving Attribute Based Access Control Model for Web Services," *Proc. Information and Communication Technologies (WICT), 2011 World Congress on,* IEEE, pp. 1223-1228, Dec. 2011, doi: 10.1109/WICT.2011.6141423.

Gang Liu received M.S. and Ph.D. degrees in computer science and technology from Xi'an Jiaotong University, Xian, China, in 2001 and 2004 respectively. He has been a faculty member of School of Computer Science and Technology at Xidian University since 2007, where he is currently an associate professor. His major research interests include embedded system, information security and trusted computing.

Huimin Song is currently a M.S. at Xidian University, Xi'an, China. Her research interests include information security, access control and security model of integrity and confident in embedded system. *The corresponding author. Email: $gliu_xd@163.com$

Can Wang is currently a M.S. at Xidian University, Xi'an, China. Her research interests include information security, access control and security model of integrity and confident in embedded system.

Runnan Zhang is currently a M.S. at Xidian University, Xi'an, China. His research interests include information security, access control and security model of integrity and confident in embedded system.

Lu Fang is currently a M.S. at Xidian University, Xi'an, China. Her research interests include information security, access control and security model of integrity and confident in embedded system.

Open Science Index, Computer and Information Engineering Vol:11, No:6, 2017 publications.waset.org/10007354/pdf