*Computers, Materials & Continua* CMC, vol.63, no.2, pp.861-871, 2020

# A Temporal Multi-Tenant RBAC Model for Collaborative Cloud Services

**Zhengtao Liu[1, *], Yi Ying[1], Yaqin Peng[1] and Jinyue Xia[2]**

**Abstract:** Multi-tenant collaboration brings the challenge to access control in cloud computing environment. Based on the multi-tenant role-based access control (MT-RBAC) model, a Temporal MT-RBAC (TMT-RBAC) model for collaborative cloud services is proposed. It adds the time constraint between trusted tenants, including usable role time constraint based on both calendar and interval time. Analysis shows that the new model strengthens the presentation ability of MT-RBAC model, achieves the finer-grained access control, reduces the management costs and enhances the security of multi-tenant collaboration in cloud computing environment.

**Keywords:** Cloud computing, multi-tenant, collaboration, access control, temporal.

## 1 Introduction

Cloud computing is a pay-on-demand model, which allows users to interact with providers with minimum management costs and services. From the configurable pool of shared computing resources, the user can obtain IT resource and services (including network bandwidth, servers, storage, applications and computing services) according to actual requirement [Mell and Grance (2011)]. Due to its unique advantages such as broadband interconnection, resource pool sharing, flexible configuration, on-demand service and charge-for-service, computing can significantly reduce the maintenance cost of computing and storage and also reduces the constraints on limited storage and computing resources for users.

A tenant is a basic logic unit of cloud computing in the management of IT resources and services. In the system, multiple tenants share the same physical resources, but logically, they are separate without affecting each other [Bezemer and Zaidman (2010)]. In the view of single tenant, it monopolizes the entire physical resources; in the level of actual deployment, each tenant has its own configuration. This single-application and multi-configuration architecture allows cloud computing resources to expand and deploy rapidly. Meanwhile, multi-tenant technology can also effectively reduce the operating costs of cloud services through dynamic resource scheduling, which is widely used by cloud service providers. The

[1] School of Computer Science and Engineering, Sanjiang University, Nanjing, 210012, China.

[2] International Business Machines Corporation, New York, USA.

[*] Corresponding Author: Zhengtao Liu. Email: lzt_jh_cn@163.com.

Received: 02 May 2019; Accepted: 23 May 2019.

introduction of tenants has brought new security problems, and one of the most important problems is how to ensure that the resources are not illegally accessed in multi-tenant environment [Feng, Qin, Yuan et al. (2015)]. Access control technology can ensure that the resources are not illegally accessed and used by restricting access rights of users to resources. In traditional computing mode, the existing access control models can do well in restricting access process and protecting resource security [Wang, Yang, Xu et al. (2015)]. However, this does not apply in a multi-tenant environment. According to the relationship between subject and object, the access in multi-tenant environment can be divided into two categories: intra-tenant access and cross-tenant access. Intra-tenant access refers to a user's access to resources within a tenant. Although the demand of the access control is similar to that of traditional access control, the access control still needs to be adapted to the dynamic cloud environment. Cross-tenant access refers to users within one tenant accessing the resources of another tenant.

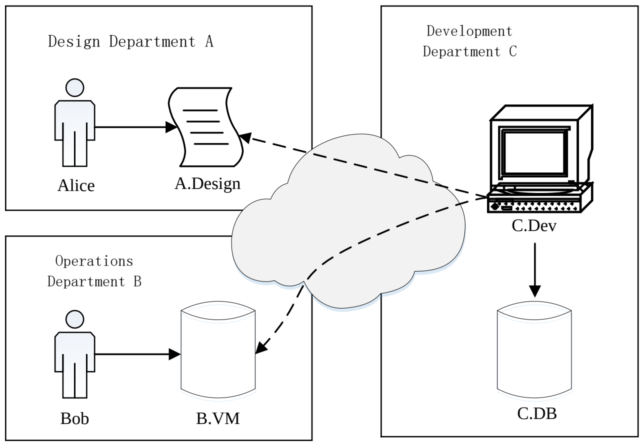

**Figure 1:** Cross-tenant cooperation scenario

As more and more services are migrated to the cloud, there is a potential cooperative relationship between tenants, which requires the constraint function of cross-tenant access control technology. In cloud computing scenarios, the cross-tenant collaboration is widespread. Fig. 1 illustrates a private cloud of an IT company. In the cloud, there are three departments: design department A, operations department B and development department C. Each department corresponds to one tenant in the cloud. Under the original structure, each department has its own responsibilities and is totally separated from the others. Section A is responsible for design resources, Section B is responsible for infrastructure resources, and Section C is responsible for software development. However, the new business requires three departments to cooperate across tenants in the private cloud. As can be seen from the Figure, there are multiple resource access requirements:

(1) Alice needs access to its department's resources A. Design to complete the design work.

(2) Bob needs to access his department's resource B.VM to complete infrastructure maintenance.

(3) The developers of department C need access to their department's database C.DB, as well as A.Design and B.VM of departments A and B, respectively.

In addition to the need for users within a tenant to access resources in other tenants, in order to improve the efficiency of resource use and ensure the priority consideration of using resources within the tenant, collaborative tenants are required to use resources at different times. Moreover, in order to prevent users of collaborative tenants from hogging resources, users within the collaborative tenants are limited to use time. Therefore, the access control technology in collaborative tenant environment needs to support time-based cross-tenant cooperation scenarios while guaranteeing tenant separation.

The remainder of this paper is organized as follows: Section 2 reviews the related works. Section 3 describes MT-RBAC, which is the basis of the proposed model. Section 4 presents the proposed TMT-RBAC model. Section 5 draws the conclusion.

## 2 State-of-the-art

In access control research, the RBAC model is one of the most widely used. The management problem is simplified in the authorization process by introducing the concept of a role and setting the role of user, which overcomes the problem of assigning access authority directly to the subject in autonomous access control. The authorization is first associated with the role, and then the role of the user is defined. The resource (object) can be accessed by the role to which the user belongs and obtained authorization of the user. Sandhu et al. [Sandhu, Coyne, Feinstein et al. (1996)] proposed a formal representation of RBAC model in 1996 (called RBAC96). On the basis of RBAC96, some researchers proposed ARBAC97 model [Sandhu, Bhamidipati and Munawer (1999)], ANSI RBAC model [Ferraiolo, Sandhu, Gavrila et al. (2001)], and so on. These models have been widely used in various application systems.

Different applications and computing resources on the same cloud server can be accessed by different tenants. Many scholars applied multi-tenant technology to the traditional access control model to generate a new access control model and enhance the access control to data. A role-based multi-tenant access control was proposed to determine the identity and the role of user by using identity management [Yang, Lai and Lin (2013)]. The application independence and data isolation were achieved by managing access rights of the tenants. A multi-tenant access control model was proposed to separate the security responsibilities of cloud service providers (CSPs) from the security responsibilities of tenants (clients) [Li, Shi, Guo et al. (2010)]. In this model, CSPs can manage the addition, deletion, management and related security issues of tenants in the cloud. In order to ensure the security of the tenant itself, the access control was managed by the tenant itself. Ngo et al. [Ngo, Demchenko and de Laat (2016)] proposed a multi-tenant attribute-based access control for cloud infrastructure services. The study focused on multi-cloud providers providing services to multiple users. An access control model was proposed and implemented for multi-cloud and multiple clients.

In order to achieve multi-tenant collaboration, Zuo et al. [Zuo, Xie, Qi et al. (2017)] provided a formal definition of a new tenant-based access control model based on administrative role-based access control (ARBAC) for multitenancy architecture and sub-tenancy architecture in service-oriented SaaS called TMS-ARBAC. Autonomous areas (AA) and AA-tree were proposed to describe the autonomy of tenants, including their isolation and sharing relationships. Authorization operations on AA and different resource-sharing strategies were defined to create and deploy the access control scheme in single-threaded apartment (STA) models. Combining the multi-tenant authentication system (MTAS) and RBAC model, Tang et al. [Tang, Sandhu and Li (2015)] proposed an administrative multi-tenant authentication system (AMTAS) model. The conditions of trust were added to MTAS, and the trust between multi-tenants was analyzed formally. On the basis of AMTAS, Tang et al. [Tang and Sandhu (2013)] proposed an MT-RBAC model. The model extended the traditional RBAC model and added two built-in components for issuers and tenants. Collaboration between different tenants was achieved through the establishment of trust relationships between tenants. Alam et al. [Alam, Malik, Akhunzada et al. (2017)] proposed a cloud resource mediation service offered by cloud service providers, which played the role of trusted third-party among its different tenants. It formally specified the resource-sharing mechanism between two different tenants in the presence of their proposed cloud resource mediation service. The correctness of permission activation and delegation mechanism among different tenants using four distinct algorithms (activation, delegation, forward revocation, and backward revocation) was also demonstrated using formal verification.

Although the above models implement access control between different tenants and across tenants, these models do not constrain the time. The restrictions are one of the main contents of RBAC research from the beginning. In order to adapt to the permission management in different situations, the expressive ability is enhanced mainly by extending the limitations of RBAC. Research on restrictions include: authorization time limits, role level restrictions, the restrictions of number of role users, and so on. The time limit is an important attribute in cloud computing environment. Due to the time factors being everywhere, users only have specific roles for a specific time period and the working mode of cloud computing is on-time billing. It is necessary to constrain access control to data by time in the cloud. Therefore, it is crucial for RBAC models to support complex time-constrained modeling. Bertino et al. [Bertino, Bonatti and Ferrari (2010)] proposed a temporal role-based access control (TRBAC) model very early on. The temporal constraints were added to RBAC.

In this paper, a new temporal multi-tenant role-based access control model based on the MT-RBAC is proposed by adding time constraints to meet the time-constrained access control requirements within tenants and across tenants.

## 3 MT-RBAC model

Three MT-RBAC models integrate three different trust relations with increasingly finer-grained constraints: tenant trust (MT-RBAC0), trustee-independent public roles (MT-RBAC1), and trustee-dependent public roles (MT-RBAC2).

The base model MT-RBAC0 requires the trustor to expose its entire role set and corresponding authorization assignments to the trustee. It consists of 6 entity components: Issuer (I), Tenant (T), User (U), Role (R), License (P), Session (S).

***Definition 1:*** T is a set of all tenants. Tenant trust relationship $TT \subseteq T \times T$ is a many-to-many relationship. For $\forall t_i, t_j \in T$, $t_i$ is the trustor, $t_j$ is the trustee, denoted as: $t_i \lhd t_j$. If $t_i$ and $t_j$ represent the same tenant, then $t_i \equiv t_j$.

$\forall t_i, t_j, t_k \in T$, the relationship of TT has the following properties:

**(1) Reflexivity:** $t_i \triangleleft t_i$ , which indicates that a tenant always trusts itself and the access within the tenant is not affected by the trust relationship.
**(2) Anti-transitive:** $t_i \triangleleft t_j \wedge t_i \triangleleft t_k \not\Rightarrow t_i \triangleleft t_k$ , which indicates that the trust relationship of any two tenants does not affect the trust relationship between the tenant and other tenants.
**(3) Anti-symmetry:** $t_i \triangleleft t_j \wedge t_j \triangleleft t_i \not\Rightarrow t_i \equiv t_j$ , which indicates that a trust relationship is one-way and independent. Mutual trust between two tenants does not mean that the two tenants are the same tenant.

If $t_i \triangleleft t_j$, the roles of $t_i$ is exposed to the issuer of $t_j$ by the issuer of $t_i$. Then, the issuer of $t_j$ can only do the following authorizations:

(1) Give the role of $t_i$ to the user of $t_j$;

(2) Take the role of $t_i$ as the senior role of $t_j$.

***Definition 2.*** MT-RBAC0 model contains the following components:

- I, T, U, R, P, S, TT are the finite sets of issuers, tenants, users, roles, permissions, sessions, and tenant trust relationships, respectively.
- $TO \subseteq T \times I$, represents a mapping of many-to-one relationships between each tenant and its issuer.
- $UO \subseteq U \times T$, represents a mapping of many-to-one relationships between each user and the tenant.
- $RO \subseteq R \times T$, represents a mapping of many-to-one relationships between each role and the tenant.
- $PO \subseteq P \times T$, represents a mapping of many-to-one relationships between each license and the tenant.
- $canUse(r:R) \rightarrow 2^T$, a function mapping a role to a set of tenants who can use the role.
- $UA \subseteq U \times R$, a many-to-many assignment relationship between a user and a role, indicating the role assigned to the user.
- $PA \subseteq P \times R$, a many-to-many assignment between a license and a role, indicating the license assigned to the role.
- $RH \subseteq R \times R$, a partial order relationship on a set of roles, called role inheritance and also referred to as "$\geq$" r, and $r_i \geq r_j$ only if $roleOwner(r_i) \equiv roleOwner(r_j) \vee roleOwner(r_i) \in canUse(r_j)$.
- $user(s:S) \rightarrow U$, a mapping between each session and a single user function that exists in the lifecycle of the session.
- $roles(s:S) \rightarrow 2^R$, a mapping of role set in the lifecycle of each session.

***Definition 3:*** MT-RBAC1 inherits all the components from MT-RBAC0 as described in Definition 2, with the following modifications:

- $P_{TI}(t:T) \rightarrow 2^R$, a new function mapping a tenant to set of roles which is the trustee-independent public role (TIPR) set of the tenant; and
- $canUse(r:R) \rightarrow 2T$ is modified to $canUse(r) = \{t\} \cup \{t \in T | t \triangleleft t_e \wedge r \in PTI(t)\}$, where (r,t)

$\in$RO

***Definition 4:*** MT-RBAC2 inherits all the components from MT-RBAC0 as described in Definition 2, with the following modifications:

- $P_{TD}(t_r,t_e:T)\rightarrow 2^R$, a new function mapping a tenant to set of roles which is the trustee-dependent public role (TDPR) set of the trustor to the trustee; and
- canUse(r:R)$\rightarrow 2^T$ is modified to canUse(r)=$\{t\} \cup \{t_e \in T| \ t \triangleleft t_e \wedge r \in P_{TD}(t_r,t_e)\}$, where (r,t) $\in$RO.

## 4 TMT-RBAC model

### *4.1 Definition*

In this paper, Time takes the concept of time interval and period.

***Definition 5.*** Time Interval. Assume TIME=$\{time_0, time_1,\ldots\}$ is an infinite set of all points on the discrete time axis. The set called time interval consists of all the time point, which is from $time_b$ to $time_e$ on the discrete time axis, and the complete set is called TD, where $time_b \leqq time_e$, $time_e$-time is the length of time, denoted as $|[time_b, time_e]|$.

***Definition 6.*** Time Period. Given calendars $C_d,C_1, \ldots, C_n$, a periodic expression is defined as PI=$\sum_{i=1}^{n} O_i \cdot C_i \triangleright r \cdot C_d$, where O1=all, $O_i \in 2^N \cup \{all\}$ and $C_i \subseteq C_i$-1 for i=2, …, n, $C_d \subseteq C_n$ and $r \in N$.

The symbol $\triangleright$ separates the first part of the expression, identifying the set of starting points of the intervals it represents, from the specification of the duration of each interval in terms of calendar $C_d$.

***Definition 7.*** PT is the time constraint set of TMT-RBAC model. PT=$\{pt_1, pt_2, \ldots, pt_n\}$, pt=($[time_b, time_e]$, pi), where the $[time_b, time_e]$ indicates the valid time of authority, and PI represents the cycle time of usage.

In order to better define the TMT-RBAC model, the functions used in the model are defined first, as shown in Tab. 1.

**Table 1:** Functions used in TMT-RBAC model

| Function | Definition | Description |
|---|---|---|
| user_o(r,time) | $R\rightarrow 2^U$ | All users with the role r in the tenant at return time |
| users_c(r,time) | $R\rightarrow 2^U$ | All users with the role r in the cross-tenant at return time |
| users(r,time) | $R\rightarrow 2^U$ | All users with the role r at return time |
| user($s_i$,time) | $S\rightarrow U$ | All users with session $S_i$ at return time |
| role($s_i$,time) | $S\rightarrow 2^R$ | All the roles of session $S_i$ at return time |
| roles_u(u,time) | $U\rightarrow 2^R$ | All the roles of user u at return time |

***Definition 8.*** The TMT-RBAC model completely inherits the characteristics of the MT-RBAC model, with its own submodels: TMT-RBAC0, TMT-RBAC1 and TMT-RBAC2. The connotations of canUse (r) of the three submodels correspond to MT-RBAC0, MT-

RBAC1 and MT-RBAC2, respectively, with the following modifications:

- Tenant trust relationship $TT \subseteq T \times T \times PT$ is a many-to-many relationship, for any $(t_i, t_j, p_{tm}) \in TT$, $t_i, t_j \in T$, $p_{tm} \in PT$, $t_i$ is a trustor and $t_j$ is a trustee, PT is a time constraint.
- canUse(r:R,pt:PT)$\rightarrow 2^T$, a function mapping a role to a set of tenants who can use the role at the time constraint of PT.
- $UA \subseteq U \times R \times PT$, $UA=\{(u,r,pt)|u \in U, r \in R, pt \in PT\}$;
- $UA=UAO \cup UAC$, of which UAO represents the distribution relationship between users and roles in the same tenant, and UAC represents the distribution relationship between users and roles across tenants. $uao \in UAO$, uao=(i.t,(t@u, t#r),pt), where i is the issuer of tenant responsible for authorization in t; u and r represent a user and a role in the tenant, respectively; pt represents the effective time; uac=(i.t,(te@u,(t#r, $pt_r$),pt'), $i,ie \in I$; (i.t,ie.te,$pt_m$)$\in$TT, roleOwner(r,$pt_r$) $\triangleleft$ t, $pt=pt_m \cap pt_r \cap pt'$.
- user_o(r,time)={u|($\exists$r'$\geq$r)(uao=(i.t,(t@u,t#r),pt)$\in$UAO$\cap$time$\in$pt(uao)}
- user_c(r,time)={ u|($\exists$r'$\geq$r)(uac=(i.t,(te@u,(t#r,$pt_r$),pt)$\in$UAO$\cap$time$\in$pt(uac)}
- users:R$\rightarrow 2^U$, users(r,time)=user_o(r,time)$\cup$user_c(r,time)
- user:S$\rightarrow$U, user($s_i$,time)={u| ua = (u,r,pt)$\in s_i \cap$time$\in$pt(ua)}
- role:S$\rightarrow 2^R$, role($s_i$,time)={r|($\exists$r'$\geq$r)(ua=(u,r,pt)$\in$UA$\cap$time$\in$pt(ua)}
- roles_u(u,time)={r'|u$\in$users(r',time), r'$\geq$r}
- valid_c(te@u,t#r,time)=$pt_m \cap pt_r \cap pt' \cap [time,\infty]$

In the TMT-RBAC model, the function of time limit mainly includes the time limit of the trust relationship and the time limit of the authorization. According to different time limits, the authorization can be divided into two kinds: permanent and time-limited. The permanent authorization means that once the user is granted permission, the user will have the permission contained in the role until the issuer revokes the role right of the user. The time-limited authorization includes time limit management of role user in the tenant and cross-tenant. The time limit management of role users in tenants is managed by the issuer: when authorizing a user role, the user is given the time limit of the role's right at the same time. The time limit management of cross-tenant is managed by the trustor and the trustee together. First, the issuer of the trustor is responsible for setting the time constraint of the trust relationship when the two parties establish a trust relationship as well as setting the time constraint of the role when the canUse(r) is set. The time constraint is only valid for the cross-tenant, but not for the user in the tenant. Based on the spirit of RBAC96, in the TMT-RBAC model, the issuer of trustee is responsible for granting the role permission of trustor to the user in the tenant as well as setting the time constraint of the role. Therefore, the time constraint of the role permission for the user is set to be $pt=pt_m \cap pt_r \cap pt'$.

In the TMT-RBAC model, the definition of canUse(r) is the same as that in MT-RBAC model, but the increased time-constraint granularity is different. The time constraint of canUse(r) of TMT-RBAC0 is for the role that can be used. The canUse(r) time constraint of TMT-RBAC1 is for the public role. The roles of the first two submodels can be used for the same time constraints for all trustees, but the TMT-RBAC2 can set different time constraints for different trustees.

### *4.2 Authorization and revocation*

Cross-tenant authorization mainly evaluates the following three aspects:

- If the trust relationship has been established between two tenants
- If the authorized role belongs to canUse(r) set
- If the authorized time is available time

The prerequisite condition defined in the ARBAC97 model is used for judgment by the authorization of TMT-RBAC model.

***Definition 9.*** The cross-tenant authorization is determined as follows:

$canAssign_c \subseteq TT \times canUse(r) \times PT$;

$assgin_c\{(t,te,pt_m), u,r, pt'\} \in canAssign_c$, then there is:

(1) $(t,te,pt_m) \in TT \cap$

(2) $u \in te@U \cap$

(3) $r \in canUse(t\#r,pt_r) \cap$

(4) $valid\_c(u,r,time) \neq \emptyset$

There are two cases of permission revocation of cross-tenant: one is the issuer voluntarily revokes the user's privilege, and the other is that the system automatically revokes the authorized role of the user based on the time limit. The former can be initiated by the issuer, for example, the issuer of the trustee revokes the role of the user, or the issuer revokes the inheritance relationship between the role of the tenant and the trustor role, or it can be initiated by the trustee. For example, by dissolving the relationship with the trustee, the trustee will revoke the rights of all users in the trusted person assigned to the role. The basis for time-based authorization revocation is the time limit of cross-tenant authorization. When the system time exceeds the maximum time limit of an authorization, the system automatically revokes the role granted by the authorized user.

***Definition 10.*** The automatic authorization revocation decision cross-tenant based on time limit.

$can\_auto\_revoke_c \subseteq U \times R \times PT$

$auto\_revoke_c \in can\_auto\_revoke_c$, then there is: $valid\_c(u,r,time)=\emptyset$.

### *4.3 Session state change pattern*

There are two kinds of changes in the state of TMT-RBAC model. The first is independent of the time change. There have been many studies of this kind of changes. The second is the change of system state, which is caused by changes of time constraints. In this paper, the second kind of changes in TMT-RBAC model will be mainly studied.

***Definition 11.*** TMT-RBAC has four states:

- *OLDS*={$s_i$|*i*∈*N*} represents the set of sessions which have been ended in the system.
- *BLOCKS*={$s_i$|*i*∈*N*} represents the set of sessions which are not terminated in the system but are blocked due to time constraints.
- *CURRENTS*={$s_i$|*i*∈*N*} represents the set of sessions which are currently in progress in the system.
- *ERRORS*={$s_i$|*i*∈*N*} represents the set of sessions which cannot be continued due to time constraints or other reasons in a system.

The intersection of two of the four sets is always empty.

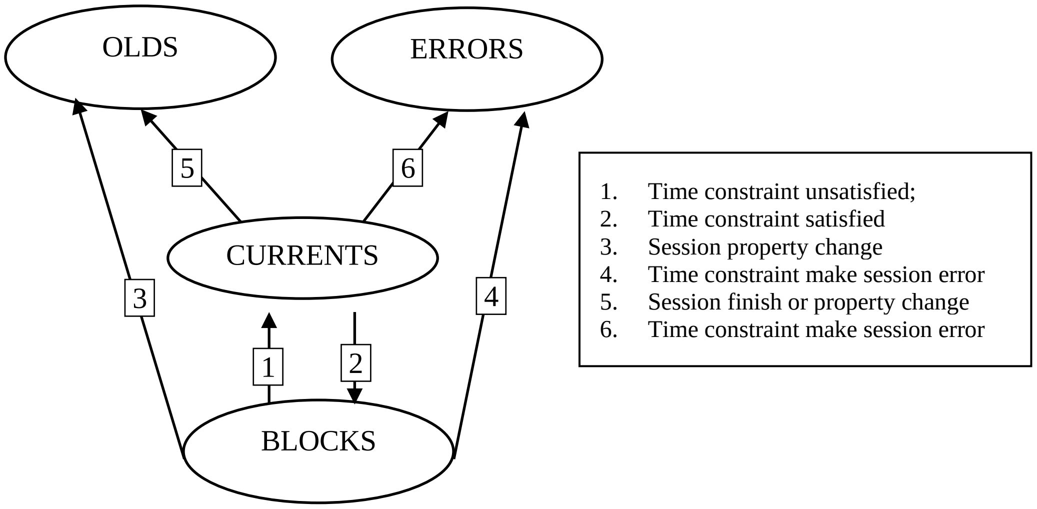
OLDS
ERRORS
CURRENTS
BLOCKS
1
2
3
4
5
6
1. Time constraint unsatisfied;
2. Time constraint satisfied
3. Session property change
4. Time constraint make session error
5. Session finish or property change
6. Time constraint make session error

**Figure 2:** Session state change pattern

In the normal state of the system, sessions will change in four sets of sessions: ERRORS, OLDS, CURRENTS and BLOCKS. When a mission is established in the system, it will be placed in CURRENTS or BLOCKS, depending on whether the time constraint of the session is meet. If the properties of the session do not change, the session may be migrated zero or more times in either CURRENTS or BLOCKS sets. It will eventually be moved to OLDS by a normal end in CURRENTS or moved to ERRORS because it cannot continue to run. However, if the properties of the session change, the session may be migrated directly from BLOCKS to ERRORS or OLDS. Fig. 2 shows the state change of the session and the type of time constraint that causes the state change.

## 5 Conclusions

In cloud computing, multi-tenant access control is incomplete and not fully secure without time limit. To address this limitation, a temporal multi-tenant role-based access control (TMT-RBAC) model was proposed, which adds the time constraint of the trustor role when two tenants establish a trust relationship. Authorization and revocation methods were defined and the changes of system space caused by increasing time constraints were discussed. Analysis indicates that the model can well achieve secure access control among tenants and cross-tenants.

In actuality, there are many kinds of time limits for cross-tenants. This paper focused on the effective use time and use cycle time in the process of authorization, without discussing the time limitation of role activation and use frequency limitation. More research is needed to analyze the fine-grained time constraints of authorization and improve the efficiency of decision-making under time constraints.

**Acknowledgement:** This work was sponsored by Qing Lan Project of JiangSu Province, The Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Nos. 17KJB520033, 17KJB520032) and National Natural Science Foundation of China (No. 61772280). The authors are grateful for the anonymous reviewers who made constructive comments and improvements.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

**Alam, Q.; Malik, S. U.; Akhunzada, A.; Choo, K. K. R.; Tabbasum, S. et al.** (2017): A cross tenant access control (CTAC) model for cloud computing: formal specification and verification. *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1259-1268.

**Bertino, E.; Bonatti, P. A.; Ferrari, E.** (2001): TRBAC: a temporal role-based access control model. *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 191-233.

**Bezemer, C. P.; Zaidman, A.** (2010): Multi-tenant SaaS applications: maintenance dream or nightmare? *Proceedings of the Joint ERCIM Workshop on Software Evolution and International Workshop on Principles of Software Evolution*, pp. 88-92.

**Feng, C. S.; Qin, Z. G.; Yuan, D.; Qing, Y.** (2015): Key techniques of access control for cloud computing. *Acta Electronica Sinica*, vol. 43, no. 2, pp. 312-319.

**Ferraiolo, D. F.; Sandhu, R.; Gavrila, S.; Kuhn, D. R.; Chandramouli, R.** (2001): Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224-274.

**Li, X. Y.; Shi, Y.; Guo, Y.; Ma, W.** (2010): Multi-tenancy based access control in cloud. *International Conference on Computational Intelligence and Software Engineering*, pp. 1-4.

**Mell, P.; Grance, T.** (2011): The NIST definition of cloud computing. *Communications of the ACM*, vol. 53, no. 6, pp. 50.

**Ngo, C.; Demchenko, Y.; de Laat, C.** (2016): Multi-tenant attribute-based access control for cloud infrastructure services. *Journal of Information Security and Applications*, vol. 27, pp. 65-84.

**Sandhu, R.; Coyne, E. J.; Feinstein, H. L.; Youman, C. E.** (1996): Role-based access control models. *Computer*, vol. 29, no. 2, pp. 38-47.

**Sandhu, R.; Bhamidipati, V.; Munawer, Q.** (1999): The ARBAC97 model for role-based administration of roles. *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 105-135.

**Tang, B.; Sandhu, R.** (2013): Cross-tenant trust models in cloud computing. *Proceedings of the 14th IEEE Information Reuse and Integration*, pp. 129-136.

**Tang, B.; Sandhu, R.; Li, Q.** (2015): Multi-tenancy authorization models for collaborative cloud services. *Concurrency and Computation: Practice & Experience*, vol. 27, no. 11, pp. 2851-2868.

**Wang, Y. D.; Yang, J. H.; Xu, C.; Ling, X.; Yang, Y.** (2015): Survey on access control technologies for cloud computing. *Journal of Software*, vol. 26, no. 5, pp. 1129-1150.

**Yang, S. J.; Lai, P. C.; Lin, J.** (2013): Design role-based multi-tenancy access control scheme for cloud services. *International Symposium on Biometrics and Security Technologies*, pp. 273-279.

**Zuo, Q.; Xie, M.; Qi, G.; Zhu, H.** (2017): Tenant-based access control model for multi-tenancy and sub-tenancy architecture in Software-as-a-Service. *Frontiers of Computer Science*, vol. 11, no. 3, pp. 465-484.