International Journal of Computational Intelligence Systems, Vol. 9, No. 1 (2016) 1-9

A Novel Role-based Access Control Model in Cloud Environments

Jun Luo [1] , Hongjun Wang [2] , Xun Gong [3] , Tianrui Li [4]

[1] Institute of Information Security, College of Mathematics, Sichuan University, Chengdu, 610054, China

E-mail: paper_xunjun@sina.com

[2] Key Lab of Cloud Computing & Intelligent Technology, Southwest Jiaotong University, Chengdu, 610031, China

E-mail: wanghongjun@swjtu.edu.cn

[3] College of Computer Science, Sichuan University, Key Lab of Cloud Computing & Intelligent Technology, Southwest Jiaotong University, Chengdu Neusoft University, Chengdu, 610054, China

E-mail: gongxun@scu.edu.cn

[4] Key Lab of Cloud Computing & Intelligent Technology, Southwest Jiaotong University, Chengdu, 610031, China

E-mail: trli@swjtu.edu.cn

Received 19 June 2015

Accepted 30 October 2015

Abstract

In Cloud environments, the relationship between resources and users is more ad hoc and dynamic. The role-based access control (RBAC) model is an appropriate access control model for Cloud environments. When using the RBAC model in Cloud environments, some new elements should be considered. This paper proposes a SAT-RBAC model (security and availability based trust relationship in RBAC) and adopts the following elements as the main factors of a trust relationship: the security state and network availability of the host used by a user, the protection state of the service providers that are related to the role. A security-based scheduling model for Cloud environments is presented. Because of the uncertainty of Cloud environments, the trust relationship is divided into three zones: the unbelievable zone, the probable believable zone and the believable zone. Bayesian method is used to estimate the trust probability distribution in the probable believable zone. This paper also provides algorithms to evaluate the values of the main elements of a trust relationship. Finally, the experiments in simulated Cloud environment based on CloudSim in PlanetLab are discussed and the results prove that the SAT-RBAC model is effective in filtering abnormal behaviors in Cloud environments.

Keywords: Access control, Cloud, Security, Availability, Trust, Bayesian.

1. Introduction

In Cloud environments, the relationship between resources and users is more ad hoc and dynamic. The resource providers for the same service are heterogeneous and distributed in different areas, so the problems of access control in Cloud environments are

much more complicated than that in general network environments.

One of appropriate access control models for Cloud environments is the role-based access control (RBAC) [1, 2] model because of the ease of authorization administration. The diversity of resources and permissions are shadowed under the special role and the user does not need to care the practical service provider. Because of the dynamic and ad hoc characteristics of Cloud environments, some new elements should be considered when using the RBAC model. Choi et al. proposed an access control scheme, combining RBAC with context awareness, to grant proper privileges to users based on their current context in ubiquitous computing environments, which is also applicable to Cloud computing environments [3]. Jang et al. provided a rule-based cloud RBAC model and its rule-based knowledge, which contains the rules managing the roles in cloud computing services and the main function of the rules is managing the role switching which occurs according to user's status, the amount of available resources or the budget limit [4]. Gomez-Baryolo et al. developed an extension of the RBAC model in multi domain environments, enabling the user to access various entities and fulfill many roles with the possibility of interacting with multiple resources, which is meaningful for Cloud environments that is full of different services [5]. Kirkpatrick et al. presented a novel framework for RBAC which prevents the user from disclosing sensitive location information when accessing a server [6]. This framework is helpful for privacy preserving that is very important in Cloud environments. Tang et al. proposed a new RBAC based access control model with two kinds of roles, in which users get credentials from owners to communicate with service provider and to get access permissions of resources [7]. Ray et al. introduced a graph-theoretic semantics of access control model to face challenges of cloud access control [8]. Zhou et al. took the spatial state, temporal state and platform trust level as contexts to present a new context-aware access control model for cloud computing [9]. Tang et al. presented a multi-tenant role-based access control model which aims to provide fine-grained authorization in collaborative cloud environments by building trust relations among tenants [10]. Banyal et al. presented the DT-BAC model which uses dynamic trust evidences of users to grant the access permission to users [11]. Some other important elements of different RBAC based models in ordinary environments also work in Cloud environments such as credentials [12, 13, 14], the history of behaviors [15, 16], the logical and device independent positions [17], the attribute and status constraints [18], and so on.

Besides the elements mentioned above, the host used by the user should be paid great attention. In open and dynamic environments such as Clouds, a normal behavior of a user may turn into a abnormal behavior when the host of the user is infected by virus or controlled by hackers, and the infection may diffuse quickly and broadly in the Cloud. So the security state of a host should be carefully evaluated before allowing any access of a user who uses the host. Besides the security state of the host, the network availability of the host also deserves attention. Too many concurrent connections and too much flow may reduce the network availability and slow the running of Cloud systems. In addition, since the providers for a same service in Clouds are multiple and various, how the servers that provide the service are protected by security measures should be evaluated before determining the access authorization. This paper integrates the security state and the network availability of a host, along with the protection state of servers into the RBAC model and adopts these components as the basis of the trust relationship between users and roles in Cloud systems.

The rest of this paper proceeds as follows: Section 2 establishes the SAT-RBAC model and provides the algorithm to determine the user-role assignment. Section 3 discusses the methods to evaluate and quantify the main elements in the model. Experiments in simulation system are provided to validate the SAT-RBAC model in Section 4. The finally section concludes this paper.

2. SAT-RBAC Model

2.1. Definition

On the base of traditional role-based access control model, a set of new elements are defined in the SAT-RBAC model as follows.

Definition 1. [*host*] A host h is a physical device used by a user to access resources, which is usually a personal computer.

Definition 2. [*host security state*] Each host h has an attribute called host security state representing its security situation, which is symbolized by λ_h.

Definition 3. [*host network availability*] A host h always consumes some network resources such as network bandwidth and TCP connections. Too much network usage indicates that there may be something abnormal. μ_h is defined as host network availability which describes the network availability of a host.

Definition 4. [*server*] A server s is a computer used by a service provider to provide different services and resources in Cloud environments. Different roles have permissions to access different resources and services provided by different servers. A server is related to a role when the role owns the permissions to access the resources and services provided by the server.

Definition 5. [*server protection state*] Each server s has an attribute named server protection state to denote how the server is protected by different security measures, which is symbolized by λ_s.

Definition 6. [*trust degree*] $T_u(r)$ is defined as a trust degree and the trust relationship between a user and resources is quantified by a trust degree. Whether a user can gain the authorization of role r is determined by the value of the trust degree.

Definition 7. [*scheduler level*] $SL(i,j)$ is defined as a scheduler level and describes the priority of server j to be scheduled to run service i.

2.2. Access control using the trust degree

The value of $T_u(r)$ is divided into three zones: (0, T_l], (T_l, T_h), [T_h, 1). (0, T_l] is the unbelievable zone. (T_l,T_h) is the probable believable zone. [T_h, 1) is the believable zone. The value of T_h is equal to the mean value of trust degrees of the successful accesses that do not result in any security event and T_l is the average of trust degrees of the accesses that result in some security events. A training sample is built on the history access records and the values of T_l and T_h are based on it. In the definitions and formulae that are provided in following sections, the value of $T_u(r)$ is never greater than 1. The policy is as follows when a user accesses resources related to a role: if the trust degree is lower than T_l, the access is rejected; if the trust degree is between T_l and T_h, the decision is made by a Bayesian method which is described in succession; if the trust degree is greater than T_h, the access is permitted and the role is authorized to the user.

Because of the uncertainty of Cloud environments, a Bayesian probability distribution [19] is used to describe the possibility that an access do not result in any security event when the trust degree is between T_l and T_h.

There are two results of an access, whether or not incurring security events. Let θ denote the probability that an access does not result in any security event. The prior probability of θ is a random variable in (0,1) which is assumed to be a uniform distribution U(0,1) whose probability density is a constant 1. When there are n accesses in which the trust degree is between T_l and T_h, the prior probability of the u successful accesses that do not result in any security event is computed in Formula (1) according to the Bayesian theorem.

$$\begin{aligned} P(\theta \mid n,u) &= \int_0^1 \theta^u (1-\theta)^{(n-u)} d\theta \\ &= \frac{(n-u)!u!}{(n+1)!} \\ &= \frac{\Gamma(u+1)\Gamma(v+1)}{\Gamma(u+v+2)} \\ (u>0, v>0, u+v=n) \end{aligned} \tag{1}$$

The Bayesian posterior distribution function $f(\theta)$ is a Beta distribution Beta($\theta|u$+1,v+1) in Formula

(2).

$$
\begin{aligned}
f(\theta) &= \frac{\theta^u(1-\theta)^{n-u}}{P(\theta \mid n,u)} \\
&= \frac{(n+1)!\theta^u(1-\theta)^{n-u}}{(n-u)!u!} \\
&= \frac{\Gamma(u+v+2)}{\Gamma(u+1)\Gamma(v+1)}\theta^u(1-\theta)^v \\
&(u>0, v>0, u+v=n)
\end{aligned} \tag{2}
$$

When there are u successful accesses which do not result in any security event in the former n accesses, the probability that the (n+1)th access in which the trust degree is between T_l and T_h is a successful one is computed in Formula (3) and whether or not an access is authorized is decided by the threshold. P_T is the threshold whose value is between 0 and 1. An access is rejected when $P(\theta|n+1,u+1)$ is lower than P_T.

$$
\begin{aligned}
&P(\theta \mid n+1, u+1) \\
&= \int_0^1 \theta f(\theta) d\theta \\
&= \int_0^1 \theta \frac{(n+1)!}{(n-u)!u!}\theta^u(1-\theta)^{n-u} d\theta \\
&= \frac{u+1}{n+2} \\
&= E(Beta(\theta \mid (u+1),(v+1))) \\
&\quad (u>0, v>0, u+v=n)
\end{aligned} \tag{3}
$$

In Cloud environments, after an access is authorized, the most appropriate server is selected to provide the service and resource that the role owns permissions to access. The most appropriate server S_j to run service v_i has the maximum value of scheduler level $SL(v_i, S_j)$ whose value is computed as follows.

$$
SL(v_i, S_j) = \frac{\lambda_{s_j}\Delta(v_i, S_j)}{\max(t_{ij}^D, t_{ij}^H)} \tag{4}
$$

where t_{ij}^D is the time which a system spends on waiting that the data is available for the running of service v_i on server S_j, and t_{ij}^H is the time which a system spends on waiting that server S_j is available for the running of service v_i. $\Delta(v_i, S_j) = t_i^E / t_{ij}^E$ is the computing performance of the server, where t_i^E is the average execution time of the service v_i on all servers, t_{ij}^E is the average execution time of service v_i on server S_j. λ_{s_j} quantifies the server protection state of server S_j, which is defined in Section 3.

2.3. Computing the trust degree

The trust degree is user-special and role-special in SAT-RBAC model. The value of a trust degree is determined by Formula (5) whose main factors are related to a user u and the role r the user wants to gain.

$$
T_u(r) = \alpha_h \lambda_h(t) \mu_h(t) \sum_1^n \omega_j \lambda_{s_j}(t) \tag{5}
$$

λ_h, μ_h and λ_s are defined in Section 2.1 and the methods to evaluate the value of λ_h, μ_h and λ_s are discussed in Section 3.

α_h represents the credit of the host whose value is determined by where the host lies. The IP address of a host has four classes: intranet one, internet one of the same ISP, internet one of other ISP and mobile internet one, whose value is 1, 3/4, 1/2 and 1/4, respectively.

ω_j is the weight of each server s_j that is related to role r and it satisfies $\sum_1^n \omega_j = 1$. In Cloud environments, the provider population for the same resource and service is ad hoc and dynamic. The value of ω_j is related to the possibility that the server s is scheduled to provide the service.

Then for each server s, the ω_j is computed as follows.

$$
\omega_j = \frac{\sum_{i=1}^m SL(v_i, S_j)}{\sum_{i=1}^m \sum_{j=1}^n SL(v_i, S_j)} \tag{6}
$$

3. Evaluating the elements

3.1. Host security states

A user is trusted only when his host can be trusted, that is, when his host is in a good security status. $T(t)$ quantifies the threat a host facing, and $V(t)$

quantifies the vulnerability of a host, $\lambda_h(t)$ is computed as follows:

$$\lambda_h(t) = \frac{1}{(1+T(t))(1+V(T))} \tag{7}$$

The following formula quantifies the threats faced by a host at time t:

$$\begin{aligned} T(t) = T(t-\Delta t \to t) + \frac{T(t-10\Delta t \to t)}{10\varepsilon} \\ + \frac{T(t-100\Delta t \to t)}{100\varepsilon^2} \end{aligned} \tag{8}$$

In Formula (8), Δt is the time window which is the sampling period of security evaluation and ε is the weighting factor whose value is between 1 and 10.

$$\begin{aligned} & T(t-\Delta t \to t) \\ & = \frac{N(t)}{N(t-\Delta t \to t)} \times \frac{C(t)}{C(t-\Delta t \to t)} \\ & \times \frac{M(t)}{M(t-\Delta t \to t)} \times \sum_{j=1}^{n} (Q_j \alpha^{D_j}) \end{aligned} \tag{9}$$

Formula (9) assumes there are n kinds of threats. Q_j is the count of events in which the threat j has taken place from time $t-\Delta t$ to t. α^{D_j} quantifies the threat j, in which α is a variable ranging from 1 to 10 whose value is determined according to the complexity of a running environment, e.g., the value of α in an open system such as Internet is much higher than in a closed system such as Intranet, and D_j is an integer ranging from 1 to 5 which represents different severity degrees of a threat. $N(t)$ is the network bandwidth usage. $C(t)$ is the CPU usage. $M(t)$ is the memory usage. Correspondingly, $N(t-\Delta t \to t)$, $C(t-\Delta t \to t)$ and $M(t-\Delta t \to t)$ represent the average usages of bandwidth, CPU and memory from time $t-\Delta t$ to t, respectively. The result of Formula (9) is higher when there are more current resource usages, because more current resource usages imply that the threats are more serious.

Formula (10) is used to quantify the vulnerability of a host:

$$\begin{aligned} V(t) = & \frac{N(t)}{1-N(t)} \times \frac{C(t)}{1-C(t)} \times \frac{M(t)}{1-M(t)} \\ & \times \sum_{j=1}^{n} (\frac{\Delta T_j}{\Delta t} \alpha^{D_j}) \end{aligned} \tag{10}$$

Formula (10) assumes that there are n kinds of vulnerability at time t. ΔT_j represents the surviving period of the vulnerability j, that is, how long the vulnerability has existed. The values and meanings of α and D_j are the same as in Formula (9). As in Formula (9), $N(t)$, $C(t)$ and $M(t)$ represent the usage of network bandwidth, CPU and memory at time t, respectively. Correspondingly, 1-$N(t)$,1-$C(t)$ and 1-$M(t)$ represent the free percentage of these resources, respectively. The vulnerability is more severe when there is more consumption of resource and the severity of the vulnerability is also directly proportional to its survival period.

3.2. Host network availability

No system owns exhaustless network resources. Usually, too much network flow or too many TCP connections indicate the possibility of a DoS attack. Two factors are taken into account to evaluate the network availability. One is the network bandwidth occupied by a host and the other is the number of the concurrent TCP connections established by the host of a user. Each host is endowed with a bandwidth quota and a connection quota. When the host h of a user u exactly consumes up its own quotas, the user gets 0.5 for μ_h. When the host consumes more or less than the quotas, the value of μ_h decreases or increases as follows:

$$\mu_h = \begin{cases} \omega_b(1-\frac{\varphi_b-\phi_b}{\varphi_b}) + \omega_c(1-\frac{\varphi_c-\phi_c}{\varphi_c}), \\ (\varphi_b \geqslant \phi_b, \varphi_c \geqslant \phi_c) \\ \omega_b(1-\frac{\varphi_b-\phi_b}{\varphi_b}) + \omega_c(1+\frac{\phi_c-\varphi_c}{\phi_c}), \\ (\varphi_b \geqslant \phi_b, \varphi_c < \phi_c) \\ \omega_b(1+\frac{\phi_b-\varphi_b}{\phi_b}) + \omega_c(1-\frac{\varphi_c-\phi_c}{\varphi_c}), \\ (\varphi_b < \phi_b, \varphi_c \geqslant \phi_c) \\ \omega_b(1+\frac{\phi_b-\varphi_b}{\phi_b}) + \omega_c(1+\frac{\phi_c-\varphi_c}{\phi_c}), \\ (\varphi_b < \phi_b, \varphi_c < \phi_c) \end{cases} \tag{11}$$

where φ_b represents the actual network bandwidth consumed by host h, φ_c is the number of the concurrent TCP connections established by host h, the

bandwidth quota is represented by ϕ_b and the quota of concurrent TCP connections number is described by ϕ_c. The weights of the two factors are denoted by ω_b and ω_c, which satisfy $\omega_b + \omega_c = 0.5$. Generally speaking, a system with more bandwidth-intensive applications gets a higher ω_b than ω_c and a system with more connection-intensive applications gets a higher ω_c than ω_b.

3.3. *Server protection state*

The security protection state of a server is quantified as follows:

$$\lambda_s(t) = \frac{1}{1+\eta_1 C(t)} \times \frac{1}{1+\eta_2 M(t)} \times R \times \frac{1}{5n} \sum_{j=1}^{n} E_j \tag{12}$$

Formula (12) assumes that there are n independent security policies used to protect the server and the validity of policy j is quantified by an integer E_j ranging from 1 to 5. R is the percentage that represents how many resources are protected by security policies. $C(t)$ and $M(t)$ are the usages of CPU and memory, respectively. η_1 and η_2 are the weighting factors. A system with more cpu-intensive applications has a higher η_1 than η_2 and a system with more memory-intensive applications has a higher η_2 than η_1. Formula (12) shows that the validity of security policy is inverse proportional to the current resource usage.

4. Experiments

Commonly, a Cloud system is constituted of three layers: the first one is the Cloud computing infrastructure layer, including physical resources such as network, disks arrays, workstations or PC servers and so on (IaaS); the second one is the Cloud computing platform layer (PaaS), including some famous projects such as Hadoop from Apache, Dynamo from Amazon, Dryad from Microsoft and Sinfonia from HP; the third one is the Cloud computing application layer (SaaS), including different application software. The SAT-RBAC model is implemented as a middleware to plug into Cloud systems and plays as an arbitrator when different Cloud applications access the resources on Cloud computing platforms.

In order to evaluate the SAT-RBAC model, a simulation system is developed which includes 50 computing nodes based on CloudSim [20] in PlanetLab [21]. CloudSim is a generalized and extensible simulation framework for Cloud computing infrastructures and PlanetLab is a novel Internet plan that provides network service and application research platforms based on the overlay networks.

Four kinds of Cloud applications are established on the computing nodes: files access, data analysis, document retrieval and mail exchange. All the Cloud resources are accessed through a portal gateway which determines whether to authorize the user according to the SAT-RBAC model and chooses the service and server pairs to schedule according to Formula (4). There are two kinds of agents: the user agent installed on each host and the server agent on each Cloud node, which take charge of evaluating the host security state and server protection state, respectively. The network usage information of each host is gathered by network sensors deployed in the system. All the information is provided to the portal gateway to determine the role-user assignment. The values of different weight factors are chosen in the following table.

Table 1. The value of different weight factors

	File access	Data analysis	Document retrieval	Mail exchange
α	6	5	5	7
ω_b	0.32	0.3	0.2	0.15
ω_c	0.18	0.2	0.3	0.35
η_1	10	20	20	20
η_2	20	15	10	10

20000 accesses from 200 different hosts which are in different security situations are simulated. The first 5000 accesses are the training sample and the value of T_l and T_h are 0.36 and 0.81, respectively. The other 15000 accesses are used to verify the SAT-RBAC model and the results are showed in Fig. 1 to Fig. 4. The gateway in SAT-RBAC model serves as

an excellent security portal and rejects most of the suspected accesses before they bring danger to the simulation system. In the experiments, the percent of permitted accesses dropped much faster in Fig. 1 than in Fig. 2 when the percent of suspected hosts rises, and the percent of security events is strictly got in control in Fig. 1 but is out of control in Fig. 2.

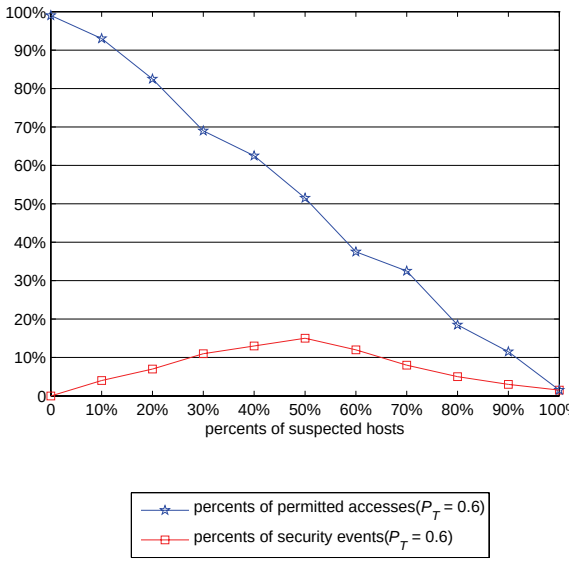

Fig. 1. Security event statistics (in SAT-RBAC model)

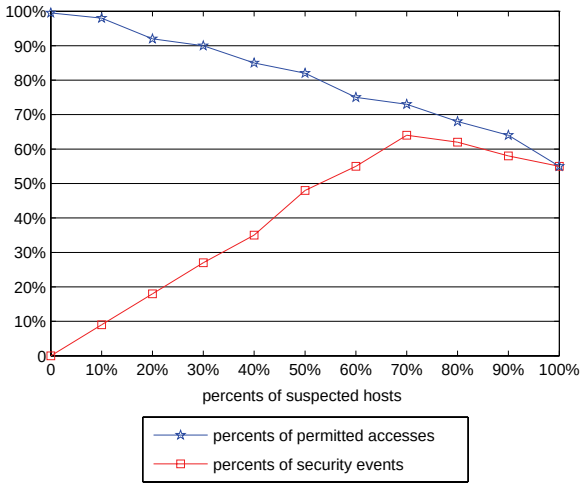

Fig. 2. Security event statistics (in RBAC model)

In the SAT-RBAC model, the access authorization decision is finally made by a Bayesian method when the trust degree is between T_l and T_h. The value of probability threshold P_T is considered. In Fig. 3, the lower value of P_T causes the higher percent of security event while the higher value of P_T results in more rejection of legal accesses. In the experiments, the most appropriate value of P_T is around 0.6.

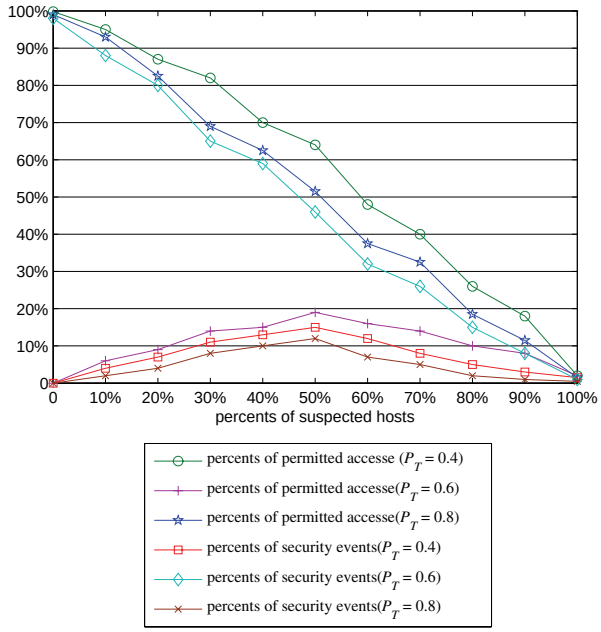

Fig. 3. Security event statistics in different P_T(SAT-RBAC)

The SAT-RBAC model also raises the success rate of legal accesses. Because most of unruly users who are accustomed to abusing the network resources are kept out of the system, the network availability of the simulation system is guaranteed. In Fig. 4, the success rate of the four types of application in the experiments is all raised, especially the success rate of the bandwidth intensive applications such as file access.

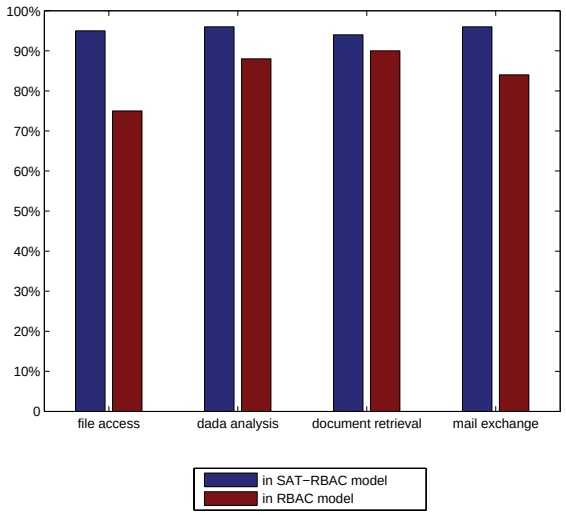

Fig. 4. Success rate of legal accesses

5. Conclusion

This paper proposes a SAT-RBAC model, which integrates host security and network availability into the traditional RBAC model to resolve the complicated access control problems in Cloud environments. In the SAT-RBAC model, whether a role is assigned to a user is determined by several elements, which include the security state and network availability of the host used by a user, the protection state of service providers that is related to a role. Mathematic formulae are provided to quantify the whole process. The result of experiments in the simulation system shows the obvious advantage of the SAT-RBAC model to filter abnormal accesses in Cloud environments.

In the future, the SAT-RBAC model will be applied and verified in the practical large Cloud system and the validity of the SAT-RBAC model will be proved in more strict mathematical forms.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions.

References

1. D. Ferraiolo, R. Sandhu, S. Gavrila, R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," *ACM Transactions on Information and Systems Security*, **4**, 224–274 (2001).
2. R. Sandhu, E. Coyne, H. Feinstein, C. Youman, "Role-Based Access Control Models," *Computer*, **29**, 38–47 (1996).
3. J. Choi, H. Jang, Y. Eom, "CA-RBAC: Context Aware RBAC Scheme in Ubiquitous Computing Environments," *Journal of Information Science and Engineering*, **26(5)**, 1801–1816 (2010)
4. E. Jang, H. J. Kim, "Rule-based Cloud RBAC Model for Flexible Resource Allocation in Cloud Computing Service," *Information – an International Interdisciplinary Journal*, **13(5)**, 1653–1666 (2010)
5. O. Gomez-Baryolo, V. Estrada-Senti, M. Lazo-Cortes, I. Garcia-Rodriguez, "RBAC Extension Model for ERP Systems in Multi domain Environments," *IEEE Latin America Transactions*, **10(5)**, 2185–2190 (2012)
6. M.S. Kirkpatrick, G. Ghinita, E. Bertino, "Privacy Preserving Enforcement of Spatially Aware RBAC," *IEEE Transaction on Dependable and Secure Computing*, **9(5)**, 627–640 (2012)
7. Z. Tang, J. Wei, A. Sallam, K. Li, R. Li, "A New RBAC Based Access Control Model for Cloud Computing ," *Lecture Notes in Computer Science*, 279–288 (2012)
8. I. Ray, I. Ray, "Trust-Based Access Control for Secure Cloud Computing," *High Performance Cloud Auditing and Applications*, 189–213 (2014)
9. Z. Zhou, L. Wu, Z. Hong, "Context-Aware Access Control Model for Cloud Computing," *International Journal of Grid and Distributed Computing*, **6(6)**, 1–12 (2013)
10. B. Tang, Q. Li, R. Sandhu, "A multi-tenant RBAC model for collaborative cloud services ," *International Conference on Privacy, Security and Trust (PST)*, 229–238 (2013)
11. R. K. Banyal, V. K. Jain, Pragya Jain, "Dynamic Trust Based Access Control Framework for Securing Multi-Cloud Environment," *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, Article No. 29 (2014)
12. N. Li, W. Winsborough, J. Mitchell, "Beyond Proof-of-Compliance: Safety and Availability Analysis in Trust Management," *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, 123–139 (2003).
13. N. Li, J. Mitchell, "RT: A Role-based Trust Management Framework," *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition*, 22–24 (2003).
14. N. Li, J. Mitchell, W. Winsborough, "Design of a Role-Based Trust-Management Framework," *Pro-*

ceedings of the 2002 IEEE Symposium on Security and Privacy, 114–130 (2002).
15. I. Ray, S. Chakraborty, "A Vector Model of Trust for Developing Trustworthy Systems," *Proceedings of the 9th European Symposium of Research in Computer Security*, 260–275 (2004).
16. S. Chakraborty, I. Ray, "TrustBAC: Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems," *Proceedings of the eleventh ACM symposium on Access control models and technologies*, 49–58 (2006).
17. L. Chen, J. Crampton, "On spatio-temporal constraints and inheritance in role-based access control," *Proceedings of the 2008 ACM symposium on Information, computer and communications security* , 20–22 (2008).
18. D. Zou, J. H. Park, T-H. Kim, X. Chen, "SH-CRBAC: Integrating Attribute and Status Constraints into the RBAC Model in Smart Home Systems," *The Computer Journal*, **52**, 861–870 (2009).
19. T. L. Griffiths, C. Kemp, J. B. Tenenbaum, "Bayesian models of cognition," *The Cambridge handbook of computational cognitive modeling* (2008).
20. T. Goyal, A. Singh, A. Agrawal, "Cloudsim: simulator for cloud computing infrastructure and modeling," *International Conference on Modelling Optimization and Computing*, **38**, 3566–3572 (2012).
21. J. M. Marques, D. Lazaro, A. A. Juan, "PlanetLab@UOC: A real lab over the Internet to experiment with distributed systems," *Computer Applications in Engineering Education*, **21(2)**, 265–275 (2013).