Journal of Intelligent & Fuzzy Systems 38 (2020) 1539–1548
DOI:10.3233/JIFS-179517
IOS Press

Distributed management of permission for access control model

Fangbo Cai*, Jingsha He, Zulfiqar Ali Zardari and SongHan
Faculty of Information Technology, Beijing University of Technology, Beijing, China

Abstract. Access control is an important mechanism to protect sensitive information and relational system resources. The traditional access control model (TACM), such as DAC, MAC, RBAC, etc., is no longer suitable for open network due to the lack of dynamic permission management. The increasing network nodes make the information storage and resource access becoming distributed. The traditional access control model has the characteristics of low adaptive ability and single deployment and application mode due to the centralized management mode. Therefore, this access control environment inevitably puts access control pressure on access control authorization. In order to overcome the shortcomings of traditional access control model, a new access control model named *DMPAC* (Distributed management of permission for access control model) is proposed in the paper. The authorization mechanism of the model has a distributed and dynamic management access permission, and all nodes covered by the model have the opportunity to participate in the execution of access and control. The model *DMPAC* provides the benefits of traditional access control models in terms of secure access and dynamic management. We also describe the framework and execution process of the model and the application of *DMPAC* in access control. At last, we will present some experimental results to show that while maintaining the effectiveness of distributed access control through the management of access permissions, *DMPAC* can achieve the performance of traditional access control models.

Keywords: Network security, access control, distributed model

1. Introduction

As an indispensable mechanism for the protection of network information and critical system resources, access control has been widely used in computer systems. With the rapid development of computer technology and information infrastructure, the network has become one of the indispensable media for interpersonal communication. More and more users store and disseminate information in the network, so the network has become the main way to obtain information resources. Access control technology is an important measure to protect network information resources. It restricts users' access to network resources, prevents illegal users from entering the system and legal users from accessing resources illegally. The goal of access control is to ensure the safe use of information and system resources [1]. To a large extent, the security of computer and network depends heavily on access control, making access control an indispensable security mechanism.

Access control was proposed by Lampson in the 1970s. Through formal description, the concept of access, Access control matrix was proposed [2]. The earliest access control model belongs to direct access control model, which mainly includes two types: discretionary access control and mandatory access control [3]. In order to meet the needs of network security, Sandhu et al. proposed the Role Based Access Control (RBAC) model, which connects the user and the authority with the Role respectively,

*Corresponding author. Fangbo Cai, Faculty of Information Technology, Beijing University of Technology, Beijing university of technology, 100 pingleyuan, chaoyang district, Beijing, China. E-mail: caifangbo@emails.bjut.edu.cn.

ISSN 1064-1246/20/$35.00 © 2020 – IOS Press and the authors. All rights reserved

making the Role become a bridge between the user and the authority. In order to meet the needs of dynamic authorization in open network environment, different access control models are proposed according to different application scenarios based on role-based access control [4]. These traditional access control models work well in closed networks, but they are not suitable for distributed and open networks. Therefore, providing dynamic and distributed access control model has been a research hot spot in recent years. Researchers have proposed task-based access control model and attribute-based access control model to support dynamic access. In the open network environment, with the arrival of cloud computing and big data, the increase of resources and the distributed characteristics of users make the traditional control model more difficult to execute. Facing the challenge of access control model, distributed theory has been applied more and more widely in the field of network and information security. Each access is controlled through an object or centralized server. Large amount of access or concurrent access in the network will bring greater access load to the central control mechanism, especially in concurrent access, which may cause access delay or unauthorized access.

To sum up, the current access control model for information protection has its own shortcomings. In the open network environment, the entities of any network domain are no longer familiar with each other, and may no longer belong to the same management institution and organization, or even belong to the same security authentication domain. Associations between users and user permissions may no longer exist. The idea of distributed management in *DMPAC* model is to store the traditional access control matrix or access control list in the central node or server into the access entities covered by the network. Distributed control model is implemented according to decentralized node agent. The access process needs cooperation of multiple entities, and the access permission judgment needs mutual restriction of multiple entities for control. It is a kind of distributed authority management mechanism that can improve the access performance to delegate users' permissions with access relations according to certain rules. Through experiments, we prove that *DMPAC* model can achieve the same access control goal as the traditional access control model and effectively protect the information resources in the network.

The rest of this paper is structured as follows: Section 1 introduces some preliminary knowledge of access control model. Section 2 studies the causes of distributed management and introduces distributed management. Section 3 introduces the proposed model and frame design in detail. In section 4, the experimental results are presented to illustrate the advantages of *DMPAC* over the existing access control model. Section 5 summarizes this paper and discusses the future work.

2. Related work

2.1. Introduction to distributed management

The distributed network management system can take advantage of the advantages of the network platform to distribute the management functions to the network instead of concentrating them in a single data center [5]. Administrators can still run the management system, and information is collected and responded to the management system by administrative agencies distributed over the network. Since the development of Distributed technology, some mature Distributed application technologies have been formed, such as Distributed Component Object Model (DCOM) [6], Common Object Request Broker Architecture(CORBA) [7], enterprise Java Bean (EJB) [8], in addition, new Web services and Grid technologies are developing rapidly and attracting more and more attention [9].

Distributed access control is an important way to protect Shared resources in the distributed management environment, which is used to restrict access to network resources and information resources in the network environment [10]. The application of distributed technology has the characteristics of autonomy and heterogeneity, which puts forward a higher requirement for the architecture of access control generality. Distributed access control strategies have diversity. Faced with different types of access control mechanisms, distributed access control has universality and can support access control strategies of multiple applications [11]. In distributed applications, different entity cooperate with each other to complete an access task [12]. In the process of interoperability, it is necessary to ensure that the access control of interoperability will not destroy the internal security mechanism of the original application, and the information access of internal security of the application is also guaranteed in the interoperability [13].

2.2. Reasons for using distributed

At present, the way of information transmission and information sharing through the network has been accepted by the broad masses, and the number of users and institutions connected to the network is increasing. The business and application carried by access control continue to expand, leading to changes in access control model, resource sharing mode, operation mode and security management method [14]. The network environment has changed from a relatively static closed network in the early days to an open network for a large number of users. The flexibility of permission allocation and security issues have also become bottlenecks restricting the development of network security and access control [15].

To sum up, the current traditional access control model is still based on the above access control mechanism. Although different control mechanisms are introduced in the study for access control decision-making and authorization to improve the effectiveness of access control and meet the requirements of different application scenarios, there are still the following deficiencies in essence. Firstly, the traditional access control model is mainly designed for the traditional computer system and closed network, without fully considering the new challenge brought by the complicated information system to the access control security. Secondly, traditional access control matrix management is used to describe the access permissions that visitors have to the visitors. The increase of users and access requests brings great complexity and security uncertainty to ensure the accuracy of permission setting and management. Third, the traditional access control mechanism is usually deployed in the server side or the visitor side, the deployment and application way is single, lack of flexibility, and centralized management is easy to become the target of network malicious attacks. Finally, the distributed authorization model of access control, how to ensure the security of distributed access control policy itself, and how to design a safe policy update method are rarely studied in this aspect.

3. The proposed *DMPAC* access control model

In order to solve the problems existing in traditional access control, in this paper, we propose a new access control model based on the idea of distributed access control authority management. This model is based on the theory of distribution, and records the access process of network nodes through the distributed management of permissions and access logs. The advantages of this model are that it can improve the security of the model, reduce the load of access judgment, resist malicious attacks and improve the efficiency of dynamic management of permissions.

3.1. Design principles

The design principle of the access control model based on distributed authorization is to change the traditional authorization access model based on single and centralized access control. The implementation process of distributed management is to store the traditional access control matrix or access control list in the central server in the form of access control list to every node covered by this access control model that participates in the access and has access relations with it. That is to say, each node stores its own accessible nodes and its own control policies, and at the same time authorizes agents for the nodes accessed by itself. The distributed access control strategy is used to replace the traditional centralized access control matrix, and each node in the network that has access relations with it manages the access control strategy of the whole network. This approach can change the traditional access control strategy centralized storage deployment mode single, poor adaptability, easy to network attack target shortcomings. At the same time, the management mode of distributed access control strategy is more secure to protect the possibility of tampering of access control strategy in the execution process, so as to play a more scientific control role in protecting the secure access of network resources. Traditional access control discrimination are stored in the central or accessed end to perform the discrimination, which will bring more policy analysis to the central node or the visitor of access control.

3.2. Access mechanism

DMPAC model changes the traditional centralized access control matrix, which is represented by access control directed graph. The directed graph $\overrightarrow{G}$ has three main components, such as access entities, access relationships, and access permissions. The collection U represents the two main elements of access control, the access subject S and the access object O, which are represented as $U = \{S, O\}$. Therefore, *DMPAC* can achieve $[S, O, E, P]$ in four aspects. The E is a set of edges in a directed graph, which

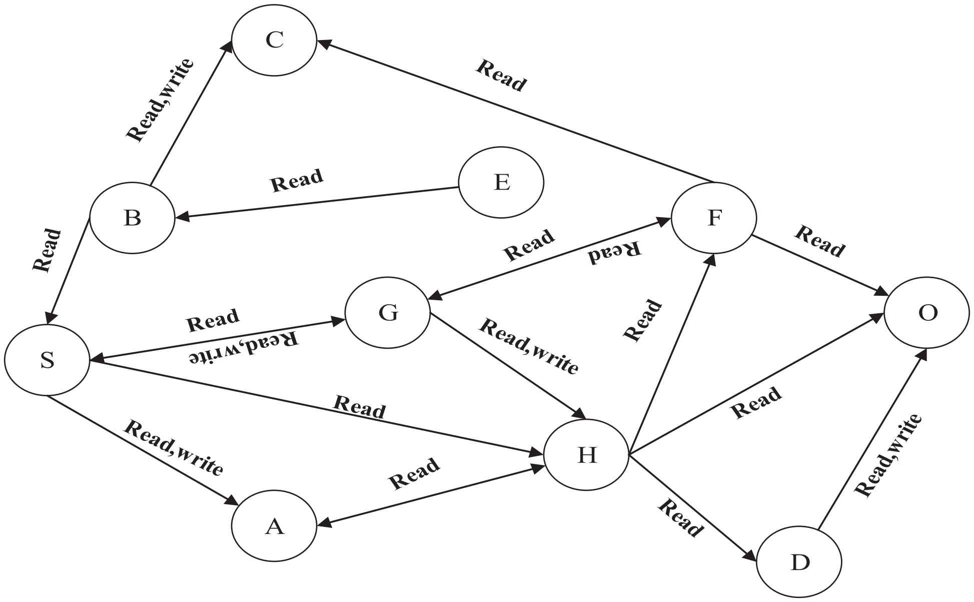

Fig. 1. Basic structure of access control directed graph.

represents the access relationship between entities. The P is a collection of permissions that represent permissions that the object allows for the subject.

Access to object content is determined by directed graph $\overrightarrow{G}$, Its vertexes are marked by entities in the model, its edges by access relations in the model, and its weights by access permissions. $\overrightarrow{G}[S, O]$ specifies the subject S access to the object O, and the weights are made up of a set of strings called access permissions. Typical permissions are $\mathrm{Re}\,ad$, $Write$. If X is a weight in a directed graph $\overrightarrow{G}$, we say that the starting entity with weighted directed edges has X access to the ending entity. For example, for the directed graph of access control in Fig. 1, the subject B has $\mathrm{Re}\,ad$ and $Write$ authority over the target object C Subject G has $\mathrm{Re}\,ad$ authority over object F.

Access control directed graph is a simplified and improved access control matrix model which can also describe the access relations presented by access control matrix. In order to better describe various access control models in the way of access control directed graph. We base the access control directed graph on a series of definitions. This model is an extension of the following six definitions, some of which can be extended or redefined based on the model in question.

Definition 1: $P\, the\, set\, of\, Permission\, (e.\,g.\, read,\, write,\, execute)$.

Definition 2: $O\, the\, set\, of\, objects\, (e.g.\, files,\, data)$

Definition 3: $S\, the\, set\, of\, Subjects\, (e.g.\, users,\, processes)$

Definition 4: $C\, the\, set\, of\, Commands$

Definition 5: $D\, the\, set\, \{grant,\, deny\}$

Definition 6: $f\, a\, function\, from\, S \times O \times P\, to\, D$

These commands provide unique methods for manipulating access control system elements, just as object-oriented methods provide unique methods for manipulating private variables of that class. S and O represent the subject and object of a visit in the model. The P is a collection of permissions that the object allows for the subject. The C is a collection of commands which are the access operations that the subject to the object. The D is the content to be accessed after considering the policy. The function f is used to determine whether a given topic has given permissions to a given object. The exact execution of this function depends on the model in question.To model the access control directed graph, we start with 1-6 as defined in this section.

Definition 7: The $\overrightarrow{G}$ is a directed graph $\overrightarrow{G} = (U, E, P)$, the U represents nodes in the graph, the E represents edges in the directed graph, and the P represents access permissions between entities, which including the entity $U = \{S, O\}$ in the access control model, where $s \subseteq S,\ o \subseteq O,\ p \subseteq P$.

Table 1
Table of entities, permissions, and session relationships

Name	Relationships	Content
PU	$PU \subseteq P \times U$	Represents the relationship between many-to-many users and permissions
US	$US \subseteq U \times S$	Represents the relationship between many-to-many users and the session
SP	$SP \subseteq S \times P$	Represents the relationship between many-to-many permissions and sessions

The commands used to create and destroy subjects are similar to those used for objects, and are omitted here and later in the model. The form of function $f(x)$ is:

$$f(s, o, p) = \begin{cases} grant & if\ p \in G[s, o] \\ deny & otherwise \end{cases}$$

Definition 8: The $\overrightarrow{G}$ can be thought of as a group where each access relationship is a subset of $\overrightarrow{G}$

Definition 9: In the access control directed graph $\overrightarrow{G}$, a subset of definition 8 that implements access is called the access control path.

Each node in the access path participates in the control of the access while performing the access. The reason for participating in the control of each node of access is to effectively resist network attacks and provide correct support for access security. When a subject S attempts to read information from the file n, the participating nodes access the $[s, n]$ during execution to see if R*ead* is one of the permissions, and if not, refuse to perform the read.

3.3. Model concept

DMPAC consists of access entity U, permission P and session S. Entity U access subject S and access object O, expressed as $U = \{S, O\}$. To simplify the model, the entity represents a network node in the model, which may assume different access roles during the access process.

The permissions collection contains all permissions that can be executed in the model. Give each access capability you have in the access environment. The nature of access control permissions depends on system setup and realistic arrangements. In general, permissions include permission for objects and permission for operations. Permissions can be applied to a single object or multiple objects, and entities and permissions are assigned in a many-to-many relationship.

Session: an entity activates a session when an access request is initiated and activates an access control directed graph stored in the entity. According to the access map of the access control directed graph, each session maps the access relationship between a pair of access entities to multiple access paths. Each session is associated with a single entity pair whose lifetime equals the lifetime of the session. In a complex network, a single entity may open multiple sessions simultaneously, each session being a sub-graph in an access control directed graph. Each session can be combined into different accesses, enabling tasks to be completed in the session.

The model is defined as follows: define the *DMPAC* model with the following components: and U, P, S (respectively represent user nodes, permissions, session)

In this model, user nodes can simultaneously support multiple sessions, and each session can be combined by different user nodes to complete the same task. The relationship of the model is shown in Table 1:

Each user node may be assigned at least one permission. As mentioned earlier, *DMPAC* treats permissions as abstract representations because permissions are real and system dependent. Our model requires that permissions apply to data and resource objects. Sessions are controlled by user nodes that, in the case of the model, can create sessions and activate access to a subset of the access paths in the directed graph. The path in the session can be changed according to the access requirements of the user node. The termination of the session is either by the subject actively or by the threshold of the length of the session.

In theory, a system security administrator can configure rules for access. Although the access idea of the *DMPAC* model is distributed management, it still supports the well-known three security principles of minimum authority, data abstraction, and responsibility separation.

3.4. Process description

Distributed access control model, each access control node of the access control model includes: access control request receiving, access control decision

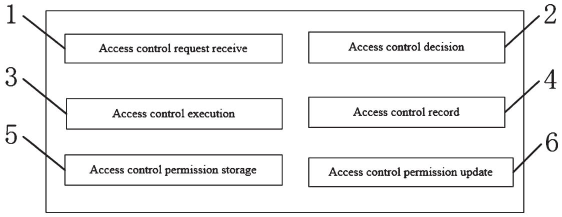
1
Access control request receive
Access control decision
2
3
Access control execution
Access control record
4
5
Access control permission storage
Access control permission update
6

Fig. 2. Access control entity structure.

making, access control execution, access control record, access control policy storage and access control policy update.

Access control request receive, used to receive access control subject access request information, access request information including access control visitors, access content and operation permissions. Access control decision is used to analyze access request information, judge whether access control is allowed or not according to the access control strategy preexisting in this node, and pass the analysis result to the access control execution module. Access control execution is used to receive the analysis results of the access control decision module. If the permission requested by the subject is less than or equal to the tolerance of self-protection of the visitor, access is allowed and access content is executed. If the visitor's request permission is greater than the visitor's tolerance of self-protection, the access is denied and the denial information is returned. The access control record module is used to record the execution record of the access control execution module. The execution record includes the access record and the rejection record. This function is convenient to realize the access tractability in network security. The access control policy storage module is used to store the access control policy of this node, the access control policy of this node and the control policy granted by the node that has access relationship with this node. Under special conditions, the access control strategy of this node can also be the access control strategy of the whole network node. The functions of each node in distributed authorization management are shown in Fig. 2:

Access method based on distributed authorization access control model, in one access, the access control subject sends a access request (subject, object, execution permission, access record). Regardless of whether the target is objected or passed through the node, the time at which the access operation is performed is recorded when the access information is received, and each access request records the execution process (access path) of each node from the subject. If the target node is saved in its own record area, if it is a path node, the judgment is performed according to the authorization level stored by the node itself. If it is determined that the application condition meets the security access requirement of the storage policy, the result of the current execution is recorded and forwarded to the next node. If it is determined that the application condition does not meet the security requirements of its storage access policy, the rejection information is returned; specifically:

Step 1. The access control subject send the access request information, and broadcasts the access request information to the neighbor node of the subject. Access request information includes subject, object and operation permission.

Step 2. The neighbor node determines, according to the received access request information, whether it is a objected that needs to be accessed by the access request information; specifically:

Access control subject, object, permission (operations performed), etc. appear in an access control request. If you receive an access control request, first check the subject and the object. First determine whether the subject meets the requirements for secure access. If it is satisfied to see who the objective is, if the objective is himself, it is judged whether the subject has the permission to access the corresponding authority. If the accessed is not the user, the node determines whether the access control request sent by the subject satisfies the security policy of the object party according to the access control policy that is present in the access control policy. If it is satisfied, the access is passed. If not satisfied, the access is denied directly.

Step 3. If the entity is the object of the subject. The object according to the list of the access permission of the user the own access control list means that all the accessible nodes and corresponding policies are included in the network node covered by the model. For example, which nodes can be accessed as a subject, which nodes can be accessed by the subject and their permission, and whether the access control request of the subject conforms to its own access control policy. If yes, provide access control subject with corresponding access content and operations; if not, return information denied access.

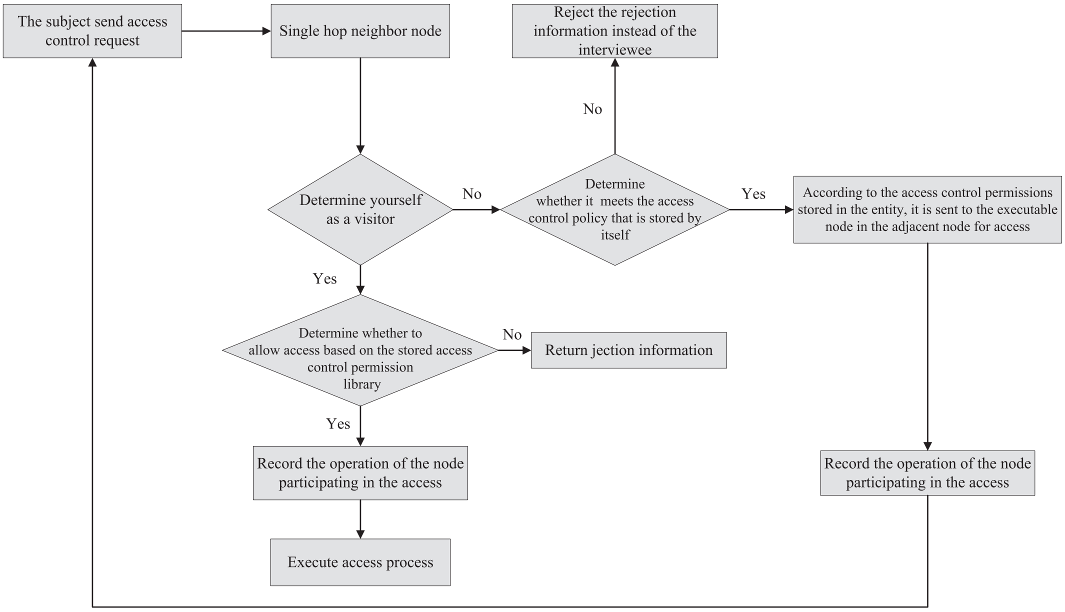
The subject send access control request
Single hop neighbor node
Reject the rejection information instead of the interviewee
No
Determine yourself as a visitor
No
Determine whether it meets the access control policy that is stored by itself
Yes
According to the access control permissions stored in the entity, it is sent to the executable node in the adjacent node for access
Yes
Determine whether to allow access based on the stored access control permission library
No
Return jection information
Yes
Record the operation of the node participating in the access
Record the operation of the node participating in the access
Execute access process

Fig. 3. *DMPAC* access control execution processs.

Step 4. If it is not the object of the access control subject (it is a passing node in an access).Then analyzing the access request information, and determining whether the access control user has the permission to access the accessed party in the access request according to the access control list stored by the node related to the self-access (which can be accessed by the path and the user can access).

Step 5. If there is permission, the access request is forwarded to the neighbor node in the access control path, and returns to step 2.

Step 6. If there is no permission, the information that is judged by the entity is returned according to the original way, and therefore the access request of the subject is rejected. The specific execution process is shown in Fig. 3:

In the distributed authorization access control model, each node provides a routing service together, and there is no so-called central server. Therefore, there is no hierarchical relationship in the architecture of this model. All nodes are equal. Each node provides services externally while using other nodes in the network. In the "client-server" access control architecture, the server acts as the decision-making center for access control. Once the central server has a problem, the execution of the access control will be in an awkward state. For the distributed authorization access control model, it has a strong security guarantee compared to the single-point failure structure of the centralized server. We can see that the nodes in the "distributed authorization" structure can be connected to other nodes, and the problem of one node does not affect the communication between other nodes, so the security guarantee is obvious.

3.5. Constraint model

Perform access control for security and convenience. We introduce constraints in the *DMPAC* model. Minimum permissions and shortest path constraints are an important aspect of *DMPAC*, which will be considered as the basis for protecting access control with minimal authorization and convenient and secure access. A common example, since *DMPAC* is the model of distributed access management and distributed control. The same subject and object can be implemented in the stored access permissions and path list with different access permissions. In such cases, duplicate and insecure access can result if there are no corresponding constraints. This is known as the principle of minimal delegation. In *DMPAC*, constraint is an effective control mechanism.

4. Experiment and analysis

Through experiments, this section verifies that *DMPAC* model can maintain the validity of the original access control model and achieve the security performance of traditional access control.

We compare the *DMPAC* model with the traditional centralized authorization access control model by accessing the simulated access platform.

4.1. Policy security analysis

In *DMPAC* model, the security factor of the policy changes with the delegated authority, because the traditional model's authority management is stable and static. This experiment discusses the influence of distributed authority management on access security. *DMPAC* is not a model change, but a change in access expression. In the basic concept, *DMPAC* model is a distributed access log book, which is a technical solution to maintain a reliable access database collectively in a decentralized way. From a data point of view, the *DMPAC* model is a distributed database in which access permissions are almost impossible to tamper with. *DMPAC* is not only embodied in the distributed storage of access policies, but also in the distributed records of access control. From a technical point of view, *DMPAC* is not a single technology, but the result of the integration of various technologies, which are combined together in a new structure to form a new access control expression, distributed access to data records and storage.

4.2. Resist attack analysis

Traditional access control models usually use different protection mechanisms to resist Dos attacks. This experiment assumes that the node security configuration and security protection level of simulated access are exactly the same. There are have 100 access entity set to randomly access a node on two experimental platforms, and the probability of setting illegal access is greater than 50%. In unit time, 500 consecutive simulated access are made to the core nodes of the two platforms. In the traditional model, the working pressure of nodes is the number of entity and the number of nodes participating in the access control. Because each access request is initiated in the traditional model, the access control entity will look for access permissions in the policy matrix list according to the permissions between the access subject and the access object, and return the result. However, in the *DMPAC* model, the node is set to be related to the node that has access relationship with itself for some or all of the authorization of its policy. After fetching the access data, in the 500 visits where the unauthorized access of 100 nodes is more than 50%, only the final number of successful access can be obtained after the permission evaluation of the core node itself. As can be seen from Fig. 4, with the increase of access times and unauthorized access times, *DMPAC* in the same experimental environment, compared with the traditional model, the *DMPAC* model will suddenly reduce the pressure of the heaviest evaluation node that can be obtained from the authority evaluation of distributed node agents, and the ability to resist DOS attacks will increase.

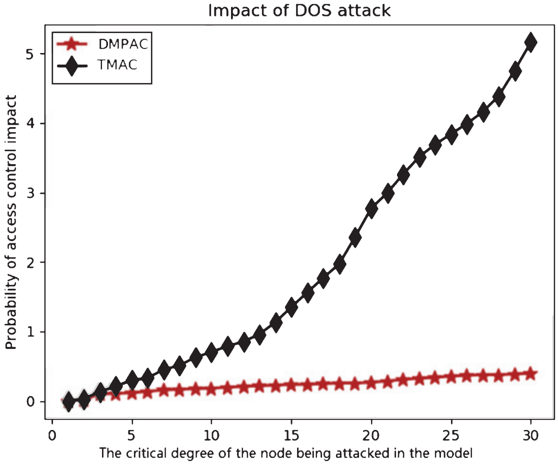

Fig. 4. The probability of a DOS attack.

4.3. Nodal failure analysis

In the traditional access control model, the centralized permission management is easy to be the target of attackers. In the model, an attack on a privileged master node will result in node failure and loss of access control capability. In the causes of node failure, we adopt two methods: selective failure and random failure. The selective failure is mainly caused by the attack on the center node of the traditional centralized access control model. The access control ability or access permission of the center node is tampered, which may lead to the access failure of the whole network. Random failure describes that network attackers attack all entities in the model at will, and the loss of access control ability of attack nodes will also affect the access of the whole model. Under the two failure modes, 100 access nodes are selected to simulate the selective failure and random failure of the nodes in the model in different

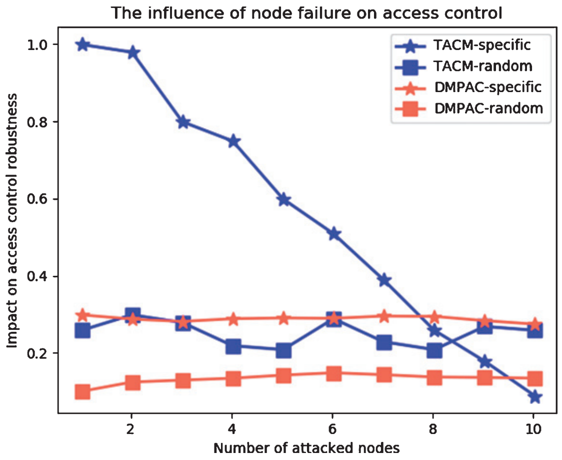

Fig. 5. Effect of node failure.

ways within unit time, and 10 data input displays are selected. The comparison effect is shown in Fig. 5.

It can be seen from this experiment that in the mode of distributed permission management of *DMPAC* model, this model has better robustness in the face of the risk of network attack and malicious tampering of strategy. In the traditional access control model, if the attacker chooses the key center management node to attack, it may cause immeasurable consequences to the access control model. Although the impact rate of the attack decreases in order according to the order of the key nodes, the model basically loses the ability of access control after the failure of the key nodes. Therefore, the *DMPAC* model proposed in this paper has better robustness in the face of malicious attacks that make its access invalid or policy malicious tampering.

5. Summary and prospect

This paper first introduces the technical means of current access control model, and analyzes the characteristics and shortcomings of current access control mechanism. This paper presents an access representation of access control directed graph. From the perspective of access control security, this paper proposes an access control model based on distributed authority management, which can effectively protect access control from network resources through distributed storage of access control authority. This paper introduces the design and implementation process of the proposed access control model based on distributed authorization, and finally verifies through experiments that the proposed access control model is more effective than the traditional access control model to protect the access control of the security and the security sharing of network resources.

In the future, we will further expand and improve the access control model based on distributed authorization, including considering the execution time cycle and execution times of access control, that is, the superposition effect of the number of visits and access cycles to the same node. We will also use business data to evaluate the performance of the model and further improve it.

References

[1] J.B.D. Joshi, E. Bertino, U. Latif, et al., A Generalized Temporal Role-Based Access Control Model[M], *A generalized temporal role based access control model for developing secure systems* **2003**, 4–23.

[2] Butler W. Lampson, Protection, *ACM SIGOPS Operating Systems Review* **8**(1) (1974), 18–24.

[3] A. Alwehaibi and M. Atay, A Rule-Based Relational XML Access Control Model in the Presence of Authorization Conflicts[M], *Information Technology - New Generations* 2018.

[4] F. Cai, N. Zhu, J. He, et al., Survey of access control models and technologies for cloud computing[J], *Cluster Computing* **2018**(2), 1–12.

[5] A. Ouaddah, A.A. Elkalam and A.A. Ouahman, Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT[M], *Europe and MENA Cooperation Advances in Information and Communication Technologies* 2017.

[6] M. Apisframeworks and M.W. Components, Distributed Component Object Model[M], *Encyclopedia of Database Systems* 2009.

[7] S. Changping, Common object request broker architecture[J], *Computer integrated manufacturing system* 1998.

[8] J. White, D.C. Schmidt and A. Gokhale, Simplifying Autonomic Enterprise Java Bean Applications Via Model-Driven Development: A Case Study[C], *International Conference on Model Driven Engineering Languages & Systems* 2005.

[9] R. Chinnici and M. Sun, Web Services Description Language (WSDL) Version 1.2[M], *Encyclopedia of Social Network Analysis and Mining* 2003.

[10] M.Y. Becker and P. Sewell, Cassandra: Distributed access control policies with tunable expressiveness[M]. 2004.

[11] K. Deinhart, Method and system for advanced role-based access control in distributed and centralized computer systems[J], *Internatl Business Mach Corp & Lt Ibm & Gt*, 1999.

[12] M. Nielsen, M. Nord and S. Bjornstad, Distributed medium-access-control protocol for an optical-packet-switched ringnetwork supporting variable-length packets[J], *Journal of Optical Networking* **4**(4) (2005), 213–225.

[13] D. Basin and T. Lodderstedt, Model driven security: From UML models to access control infrastructures[J], *Acm Transactions on Software Engineering & Methodology* **15** (2006), 39–91.

[14] S. Ruj and A. Nayak, A decentralized security framework for data aggregation and access control in smart grids[J], *IEEE Transactions on Smart Grid* **4** (2013), 196–205.

[15] M. Röscheisen and T. Winograd, A network-centric design for relationship-based security and access control[M]. 1997.