SCIENCE CHINA
Information Sciences Manuscript

• **RESEARCH PAPER** • January 2014, Vol. 57 xxxxxx:1–xxxxxx:15
doi: xxxxxxxxxxxxxx

A Novel Cyberspace-Oriented Access Control Model

Fenghua Li, Yanchao Wang, Rongna Xie, Fangfang Shan& Jinbo Xiong

State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China

Received ; accepted

Abstract With the developments of mobile communication, networks and information technology, many new information service patterns and dissemination modes emerge with some security and privacy threats in access control, i.e., the ownership of data is separated from the administration of them, secondary/mutiple information distribution etc. Existing access control models, which are always proposed for some specific scenarios, are hardly to achieve fine-grained and adaptive access control. In this paper, we propose a novel Cyberspace-oriented Access Control model, termed as CoAC, which avoids the aforementioned threats by comprehensively considering some vital factors, such as the access requesting entity, general tense, access point, resource, device, networks, internet-based interactive graph and chain of resource transmission. By appropriately adjusting these factors, CoAC covers most of typical access control models and fulfills the requirements of new information service patterns and dissemination modes. We also present the administrative model of our proposed CoAC model and formally describe the administrative functions and methods used in the administrative model by utilizing Z-notation. Our CoAC is flexible and scalable, it can be further refined and expanded to figure out new opportunities and challenges in the upcoming access control techniques.

Keywords Cyberspace Security, Access Control, Administrative Scene, Information Service Pattern, Information Dissemination Mode

Citation . Title. Sci China Inf Sci, 2014, 57: xxxxxx(15), doi: xxxxxxxxxxxxxx

1 Introduction

The rapid developments of mobile communication, networks and information technology, especially the emerging technologies such as cloud computing and big data, cause a set of new information service patterns, novel information dissemination modes and at the same time launch a revolution on people's social activities. They accelerate the information service patterns and dissemination modes into a novel pattern, accessing resources in "system of system" via "network of network". Through this way, "people", "machines" and "things" will be combined together by the ubiquitous networks, we can easily access resources anytime, anywhere with any devices and via any internet access available. However, every coin has two sides, these comfortable conveniences come at cost of the system security and our privacy. Thus, how to effectively manage and control users' accessing behaviors in the information systems becomes a vital problem.

As a core technology of information system security, access control can monitor the accessing activities effectively, guarantee the rights and permissions for the legitimate users while preventing unauthorized users from accessing system resources. Researchers have proposed a number of access control

models for different scenarios, including the Discretionary Access Control (DAC)[1],Mandatory Access Control (MAC)[2], Role-Based Access Control (RBAC)[3], Task-based authorization controls (TBAC)[4], Distributed-Oriented Access Control [5, 6], space and time-associated access control [7, 8], attribute-based access control [9, 10] and action-based access control [11], etc. To solve the new problems brought by the well developed cloud computing, such as the ownership of the data is separated from the administration, multi-tenant, etc., researchers proposed some cryptographic tools-based access control models [12, 13].

However, the aforementioned access control models are always proposed for some specific scenarios, which are hardly to achieve fine-grained and adaptive access control. In this paper, we propose a novel Cyberspace-oriented Access Control model, termed as CoAC, with comprehensively considering the access requesting entity, general tense, access point, resource, device and networks, etc. CoAC covers most of existing access control models, the characteristics and its novelties can be concluded as follows.

1) We propose CoAC, taking some key factors in the access control into consideration, including the access requesting entity, general tense, access point, resource, device, networks, internet-based interactive graph and chain of resource transmission. CoAC can guarantee fine-grained and adaptively access control, and achieve flexible information sharing in the ubiquitous network environments.

2) CoAC covers most of existing access control models. It can be utilized in different scenarios, and describe some popular access control models like DAC, MAC, RBAC and ABAC, etc., by adjusting the factors involved in our CoAC.

3) We present the definition of administrative scene and the construction of the administrative model. We describe the control relationship between the ARE (access requesting entity)-ADSC (administrative scene) and ADSC (administrative scene)-ADP (administrative permission). We also formally define the management function used in the scenes and show the administrative methods in CoAC.

The rest of this paper is organized as follows. In Section 2, we review related work. Section 3 and 4 present our CoAC model and the related administrative model, respectively. Section 5 presents how to use our model to describe existing models. We draw the conclusions in Section 6.

2 Related work

Existing access control models enable authorized user to gain reasonable access while preventing unauthorized user from accessing system resource by effectively managing and controlling user's accessing behaviors on the information resources. From the 1970s, the access control technology have gone through the following stages of development.

To solve the administrative problem of data sharing and access control on the mainframe, researchers have proposed Discretionary Access Control (DAC) model and Mandatory Access Control (MAC) model. The basic idea of DAC [1] is that the object owner have full rights to transfer the access permission to another owner. Authors in [2] proposed MAC. The basic idea is that security administrator of information system, which is marked with permissions, attaches specific secret level tag to every access object in system, and transfers the access level to every user. These are called security property of subject and object respectively, assigned by security administrator [14]. DAC can provide flexible access control policy, but its security is poor. In MAC, the information flow goes one way through setting the security level to users and data. However, MAC has its own limitations such as heavy workload, low efficiency and less flexibility.

With the development of information system and LAN technology, researchers proposed Role-Based Access Control (RBAC) model and Task-Based Access Control (TBAC) based on DAC and MAC. RBAC [3, 15, 16, 17, 18] gives each user a role, which is mapped to the related permission to implement access policy. Each particular user has multiple roles, and each role can be granted to multiple users simultaneously. It simplifies permission administration under various environments. TBAC, an application and business oriented access control model, constructs security model and implements security mechanism from the task point of view. TBAC provides dynamic and real-time security administration [4, 19, 20] during task processing. Besides, as a context-sensitive access control model, TBAC can perform fine-

grained security policies on either tasks in a single workflow or different workflows. It can be applied to workflows, security administrations, information processing as well as distributed processing.

With the rapid development of internet technology, various access control models are proposed, including distributed environment and cross-domain oriented access control fitting into B/S structure [5, 6], access control associated with space and time [7, 8, 21, 22, 23], attribute-based access control [9, 10, 24, 25] and action-based access control [11]. Authors in [5] proposed Distributed Role-Based Access Control (DRBAC) based on RBAC, DRBAC provides cross-domain cooperation with a extensible distributed trust management and access control mechanism. However, there may exist several security risks when performing role mappings. Bertino *et al.* [7, 8] extended space and time-based RBAC model, they constructed GEO-RBAC access control model. It uses spatial entity to describe objects, user locations and geographical-position-based roles. It also supports the physical locations as well as logical locations. Authors in [21] proposed TRBAC by adding temporal constraints to the RBAC model. Wang *et al.* [22] achieved dynamic access control based on roles by defining a novel tense hierarchy mechanism. Xu *et al.*[23] extended single subject to multiple subjects in TRBAC and fit for the complex network environments. Attribute-Based Access Control (ABAC) [9, 10, 24, 25] solved problems of fine-grained access control and large-scale user dynamic extension in complex information system and provided open networks a desired access control model. Li *et al.*[11] proposed a Action-Based Access Control (ABAC) model and discussed the relations among role, tense and environment in ABAC. It also provides the definitions of administration action and ABAC administration model. Compared with existing models, ABAC model is more suitable for the access control in information system, which supports mobile computing under network environment.

With the rapid development of cloud computing and cloud application, people upload their personal data to the cloud. Users can enjoy the convenience provided by the service providers. However, since the ownership of data is separated from the administration of them, users lose control on their personal data in the unpredictable cloud, who may abuse them later. Traditional access control based on policy management cannot guarantee that data would be accessed according to policies drafted by data owner. Encrypting data with Specific encryption algorithm is a basic method to ensure data confidentiality in cryptology. Access control mechanism based on policies and storage of ciphertext on cloud servers are mutually complementary. Then access control mechanisms based on cryptographic algorithm were proposed. Access control based on policies can be classified into timed-release encryption (TRE), identity based encryption (IBE), attribute based encryption (ABE) and etc. May[26] firstly proposed TRE in 1993. Rivest *et al.*[27] did further research and proposed TRE encryption scheme based on computational complexity and the trusted third party agent. Cathalo *et al.*[28] presented an efficient and non-interactive TRE encryption scheme. Existing TRE scheme mainly use trusted time server to broadcast time trapdoor, but Paterson *et al.*[29] proposed a new Time-Specific Encryption (TSE) scheme. Users can get the time trapdoor in a time interval to avoid missing the time trapdoor. Considering huge number of users and the uncertainty in cloud computing, Zhou *et al.*[30] presented a role-based encryption cloud security storage system. In this encryption system, only users with specific role can decrypt ciphertext and get the corresponding plaintext. So far the implementation mechanism of ABE includes basic ABE, KP-ABE and CP-ABE. Basic ABE can be only used in application scene with simple policy. KP-ABE and CP-ABE can support complex policies which are widely used in fine-grained data share and management. In cloud storage system the service providers always come from different trust domains. Wang *et al.*[31] proposed a fine-grained hierarchical attribute-based access control mechanism (HABE) and the constraint conditions. Hong *et al.*[32] proposed an ciphertext-based access control of cloud storage which greatly reduce the overload of permission administration. Cheng *et al.*[33] proposed a re-encrypt optimization scheme based on data segment to avoid re-encrypting the whole data once data changes. Ye *et al.*[34] proposed an ABE scheme supporting user revocation. In order to solve the fine-grained attribute revocation problem, Wang *et al.*[35] presented a CP-ABE scheme supporting fully fine-grained attribute revocation. In cloud and multi-cloud computing environment, there are always multiple authorities. Chase *et al.*[36] proposed the multi-authority ABE scheme. There are multiple authorities and a trusted central authority in this scheme. Every user would get a global identifier (GID) that will be embedded in the keys of users. This

can efficiently prevent collusion attacks from unauthorized users. Then, Chase *et al.*[12] presented a practical multi-authority KP-ABE scheme which could protect privacy of the tenants and did not rely on the trusted central authority. Liu *et al.*[13] proposed a multi-owner data sharing scheme for dynamic group in the cloud. It leveraged group signature and dynamic broadcast encryption techniques to achieve data sharing anonymously among group users.

3 Construction of our CoAC Model

In this section, we first present the system model. Then we present some basic definitions, its hierarchy structures and their formal description. Finally, we propose our cyberspace-oriented access control model.

3.1 Our System Model

As shown in Fig. 1, the system model of our CoAC model includes three entities, they are Access Requesting Entity (ARE), general network and resources, respectively.

Access Requesting Entity: ARE denotes the initiator of resource access. It sends the accessing request to the resource server through the network/general network. When assigning access permission to ARE, the resource server needs to consider many access control factors such as the device, general temporal status, access point and resource, etc.

Network/general network: it denotes the carriers of information dissemination during information accessing, which can be any network form connected, such as mobile internet and wired network. This concept also includes the network form connected by the traditional dissemination tools, for example, disk, flash, papers and so on.

Resource: it refers to the object and the related properties in the access control models.

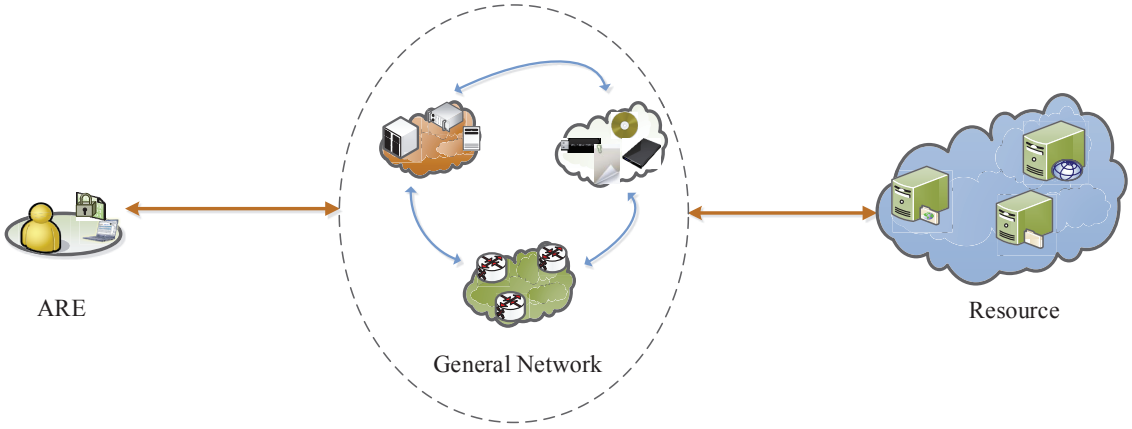

Figure 1 System Model of CoAC

In our CoAC model, the ARE issues and sends the information accessing request to the resource server via general networks. The resource server accepts the request and reply with the required resource back to the ARE once the employed device, general tense, access point can be matched with the pre-defined strategies, and vise versa.

3.2 Some Basic Definitions

Definition 1. Access requesting entity (ARE) refers to the initiator of an information accessing request. It is denoted by $q = \langle u, a, r \rangle$, where u represents the user's unique identity, a means the unique identity of an access agent that can be either a device, process or another user, r indicates the role of a particular user. We define the user set, access agent and role by $U = \{u_i \,|\, i \in \mathbb{N}^*\}$, $A = \{a_i | \, i \in \mathbb{N}^*\}$ and $R = \{r_i \,|\, i \in \mathbb{N}^*\}$, respectively. Thus, the ARE set is defined as $Q = \{q = \langle u, a, r \rangle \,|\, u \in U, a \in A, r \in R\}$.

Note that, we use $q = \langle u, \cdot, \cdot \rangle$ to describe all the information accessing requests queried by user u, $q = \langle u, \cdot, r \rangle$ represents all the information accessing requests queried by user u, with the role r. For simplicity, we ignore the similar descriptions of the other cases in details.

Definition 2. General tense means all the tense-related information when the ARE accessing resources. It is denoted by $T = \{t = \langle interval, period, duration \rangle \mid interval \in 2^{T^{IN}}, period \in \mathbb{R}^+, duration \in \mathbb{R}^+\}$, where $interval \in 2^{T^{IN}}$ refers to the starting and ending time of a particular access, and $T^{IN} = \{[begin_i, end_i] \mid i \in \mathbb{N}^*\}$. *period* represents the life cycle of a period, and *duration* represents the access duration. Note that, we have $t = \langle [begin_1, end_1], [begin_2, end]_2, \cdots, [begin_{|T^{IN}|}, end_{|T^{IN}|}] \rangle$, if $|T^{IN}| \neq 1$. Otherwise, we have $t = \langle [begin, end], period, duration \rangle$, if $|T^{IN}| = 1$.

Definition 3. Access point refers to the spatial location and network identity when an ARE accessing the network system and querying the information accessing request for the first time. The accessing system distinguishes different accesses based on the unique network identity. The access point is denoted by $l = \langle l^{SPID}, l^{NETID} \rangle$, where $l^{SPID} = \langle x, y, z \rangle \in L^{SPID}$ means the location in three-dimensional coordinate system, i.e., x, y and z may be the latitude, longitude and hight. $L^{NETID} = \{imei, bs, nid, port, ip, domian, \cdots\}$ represents the unique identity of network access, where *imei* means the international mobile equipment identity, *bs* indicates the unique identity of the base station, *nid* is the network ID, *mac* refers to the MAC address, *port* is the port interface. *ip* means the IP address, *domain* means the domain name, etc. Then the access point set is defined as $L = \{l = \langle l^{SPID}, l^{NETID} \rangle \mid l^{SPID} \in L^{SPID}, l^{NETID} \in 2^{L^{NETID}}\}$, where L^{SPID} and L^{NETID} represent the spatial location and network identity, respectively.

Definition 4. Resource refers to the accessed object and the relevant properties, it is denoted by $o = \langle c^O, g^O, s^O \rangle$, where $c^O \in C^O$, $g^O \in G^O$ and $s^O \in S^O$ refer to content, general property and security property of resources, respectively. The general property set of resources is defined as $G^O = \{g^O = \langle g^{O_{SORT}}, g^{O_{SOURCE}}, g^{O_{SIZE}}, g^{O_{TIME}}, \cdots \rangle \mid g^{O_{SORT}} \in G^{O_{SORT}}, g^{O_{SOURCE}} \in G^{O_{SOURCE}}, g^{O_{SIZE}} \in G^{O_{SIZE}}, g^{O_{TIME}} \in G^{O_{TIME}}, \cdots\}$. In this set, $G^{O_{SORT}}$ represents types of resource, denoted as $G^{O_{SORT}} = \{database, file, web, \cdots\}$, which are database, file and web, etc. $G^{O_{SOURCE}}$ denotes the source of information, denoted as $G^{O_{SOURCE}} = \{creat, forward, reconstruct, \cdots\}$, which means the creation, forwarding, reconstruction and etc. $G^{O_{SIZE}}$ denotes the size of resource. $G^{O_{TIME}}$ represents the time property of resource as mentioned in Def. 2. The security property set of resource indicates the security-related permissions on a particular resource, such as permissions of executing, forwarding and destroying etc. It is defined as $S^O = \{s^O = \langle s^{O_{OP}}, s^{O_{DIS}}, s^{O_{DE}}, s^{O_{SEC}}, s^{O_{ENC}}, \cdots \rangle \mid s^{O_{OP}} \in 2^{S^{O_{OP}}}, s^{O_{DIS}} \in 2^{S^{O_{DIS}}}, s^{O_{DE}} \in S^{O_{DE}}, s^{O_{SEC}} \in S^{O_{SEC}}, s^{O_{ENC}} \in S^{O_{ENC}}, \cdots\}$. In this set, $S^{O_{OP}}$ represents the allowed operations, denoted by $S^{O_{OP}} = \{read, edit, copy, forward, print, revoke, \cdots\}$. $S^{O_{DIS}}$ means the resource distribution methods, denoted by $S^{O_{DIS}} = \{u\text{-}disk, cd/dvd, encryption, network, \cdots\}$. $S^{O_{DE}}$ represents resource destroy methods, denoted by $S^{O_{DE}} = \{logicaldel, physicaldel, \cdots\}$. $S^{O_{SEC}}$ refers to the security level of resource, denoted by $S^{O_{SEC}} = \{top\ secret, secret, \cdots\}$. $S^{O_{ENC}}$ indicates the encryption methods, denoted by $S^{O_{ENC}} = \{3DES, RSA, ECC, AES, SM2/3/4, \cdots\}$. Then we define the set of resources as $O = \{o = \langle c^O, g^O, s^O \rangle \mid o^O \in c^O, g^O \in G^O, s^O \in S^O\}$.

Definition 5. Device refers to the equipment, which is used to access resources by the ARE. We define the composition characteristics and relevant properties as $d = \langle g^D, s^D, t \rangle$, where g^D, s^D and t represent the general property, security property and general tense of device, respectively. They are denoted by $D = \{d = \langle g^D, s^D, t \rangle \mid g^D \in 2^{G^D}, s^D \in 2^{S^D}, t \in T\}$. In this set, G^D represents the general property set, denoted by $G^D = \{cpu, os, interface, memory, disk, app, \cdots\}$, where *cpu*, *os*, *interface*, *memory*, *disk* and *app* indicates the CPU, operating system, interface, main memory, disk and applications in the device, respectively. S^D represents the security property set, denoted by $S^D = \{[mincoe, maxcoe], security\ domain, securitylevel, securitysoftmodule, securityhardmodule, \cdots\}$, where *mincoe* and *maxcoe* indicate minimum and maximum risk tolerance coefficient, *securitydomain* means security domain, *securitylevel* refers to security level, *securitysoftmodule* shows security software module, and *securityhardmodule* represents security hardware module, etc. T represents the general tense set, which covers all the tense-related properties of the device and similar as the definition in Def. 2.

Definition 6. Network refers to information transmission carrier that is the set of different networks. We define the set of network as a directed property graph $NG = (V, E)$, where vertex of a graph represents a device or a subnetwork in the network, and edge of a graph denotes connected property and security property of vertexes.

Vertex is denoted by $v = \langle n^V, g^V, s^V \rangle$, where $n^V \in N^V$, $g^V \in G^V$ and $s^V \in S^V$ represent name, general property and security property of the vertex v, respectively.

The general property set of vertex is defined as $G^V = \{g^V = \langle g^{V_{NT}}, g^{V_{IO}}, g^{V_{NP}} \rangle \,|\, g^{V_{NT}} \in G^{V_{NT}}, g^{V_{IO}} \in G^{V_{IO}}, g^{V_{NP}} \in 2^{G^{V_{NP}}}\}$. In this set, $G^{V_{NT}}$ represents set of network types, denoted by $G^{V_{NT}} = \{lan, wan, wlan, \cdots\}$. $G^{V_{IO}}$ means set of vertex types, denoted by $G^{V_{IO}} = \{in, out, inout, interior, \cdots\}$. $G^{V_{NP}}$ refers to set of network protocols, denoted by $G^{V_{NP}} = \{$*TCP/IP*, *Bluetooth*, *802.11a/b/g/h*, *ISO1989*, *CDMA2000/WCDMA/TD-SCDMA*, *LTE*, $\cdots\}$.

S^V indicates vertex's security property set. When a vertex represents a device, S^V is similar as the Definition S^D in 5. When a vertex denotes a network, we define its security property set as $S^V = \{s^V = \langle s^{V_{CON}}, s^{V_{ENC}}, s^{V_{PT}} \rangle \,|\, s^{V_{CON}} \in S^{V_{CON}}, s^{V_{ENC}} \in 2^{S^{V_{ENC}}}, s^{V_{PT}} \in 2^{S^{V_{PT}}}\}$. In this set, $S^{V_{CON}}$ represents manage and control information. $S^{V_{ENC}}$ means set of encryption modes, denoted by $S^{V_{ENC}} = \{$*3DES*, *RSA*, *ECC*, *AES*, *SM2/3/4*, $\cdots\}$. $S^{V_{PT}}$ refers to set of security protocols, denoted by $S^{V_{PT}} = \{$*SSL*, *SSH*, *HTTPS*, *MANCONFIRM*, $\cdots\}$.

Thus we define Vertex set as $V = \{v_i = \langle n_i^V, g_i^V, s_i^V \rangle \,|\, n_i^V \in N^V, g_i^V \in G^V, s_i^V \in S^V, i \in \mathbb{N}^*\}$.

Edge is denoted by $e = \langle v_m, v_m, g^E, s^E \rangle$, where $v_m \in V$, $v_n \in V$, $g^E \in G^E$ and $s^E \in S^E$ represent the starting vertex, the ending vertex, connected property and security property of edge e, respectively.

We define edges' general property set as $G^E = \{g^E = \langle g^{E_M}, g^{E_{NP}} \rangle \,|\, g^{E_M} \in 2^{G^{E_M}}, g^{E_{NP}} \in 2^{G^{E_{NP}}}\}$, where $G^{E_M} = \{$*bandwidth*, *QoS*, *hop*, *delay*, $\cdots\}$ represents set of performance characteristics, and $G^{E_{NP}} = \{$*TCP/IP*, *Bluetooth*, *802.11a/b/g/h*, *ISO1989*, *CDMA2000/WCDMA/TD-SCDMA*, *LTE*, $\cdots\}$ denotes set of network protocols.

The set of edges' security properties is denoted by $S^E = \{s^E = \langle s^{E_{ENC}}, s^{E_{PR}} \rangle \,|\, s^{E_{ENC}} \in S^{E_{ENC}}, s^{E_{PR}} \in S^{E_{PR}}\}$, where $S^{E_{ENC}} = \{$*3DES*, *RSA*, *ECC*, *AES*, *SM2/3/4*, $\cdots\}$ means set of encryption mode, and $S^{E_{ENC}} = \{$*SSL*, *SSH*, *HTTPS*, *MANCONFIRM*, $\cdots\}$ indicates set of security protocols.

Then we define edge set as $E = \{e_i = \langle v_{i_m}, v_{i_n}, g_i^E, s_i^E \rangle \,|\, v_{i_m}, v_{i_n} \in V, g_i^E \in G^E, s_i^E \in S^E, i, i_m, i_n \in \mathbb{N}^*\}$.

Finally, Network is denoted as a directed property graph $NG = (V, E)$.

Definition 7. Internet-based interactive graph refers to the network sub-figure constructed by the connected path between two arbitrary vertexes. We define it based on exchange action and chain of internet transmission.

Exchange action consists of two connected and adjacent nodes and the corresponding edge. If for vertexes $v, w \in NG(V, E)$ there exits a edge $e = \langle v, w, g^E, s^E \rangle \in E$, we define the exchange action between v and w as $N = \langle v \to w, t \rangle$, where $v \in V$, $w \in W$ and $t \in T$ represent vertexes and tense property, respectively. The set of all exchange actions in network graph is defined by $N^G = \{N_i = \langle v_i \to w_i, t_i \rangle \,|\, \forall e_i = \langle v_i, w_i, g_i^E, s_i^E \rangle \in E, i \in \mathbb{N}^*, t_i \in T\}$.

Chain of internet transmission indicates the set of ordered exchange actions. We suppose that u and v represent information sender and information receiver. Thus the chain of internet transmission between u and v is defined by $N(v, w) = \{N_i(v, w) = \langle N_{i_1}, N_{i_2}, \cdots, N_{i_j}, \cdots, N_{i_k} \rangle \,|\, i, k, i_j \in \mathbb{N}^*, 1 < j \leqslant k, v_{i_1} = v, w_{i_k} = w, t_{i_j} \in T, N_{i_j} = \langle v_{i_j}, w_{i_j}, t_{i_j} \rangle \in N^G, where\, for\, \forall 1 < j \leqslant k, w_{i_j} = v_{i_j+1}\}$. Then we denote the set of all exchange actions in internet transmission chain as $N^C(v, w) = \{N_{i_j} = \langle v_{i_j}, w_{i_j}, t_{i_j} \rangle \,|\, \forall N_i(v, w) = \langle N_{i_1}, N_{i_2}, \cdots, N_{i_j}, \cdots, N_{i_k} \rangle \in N(v, w), i, k, i_j \in \mathbb{N}^*, 1 < j \leqslant k\}$.

Since there are unidirectional edges and bidirectional edges in directed property graph $NG = (V, E)$, interactive graph, consisting of all connected information transmission chains between v and w, includes any combination of unidirectional edges and bidirectional edges. It is defined as $NG_N = (V(v, w), E(v, w))$, where $V(v, w) = \{v_{i_j} = \langle n_{i_j}^V, g_{i_j}^V, s_{i_j}^V \rangle \,|\, i_j \in \mathbb{N}^*, \forall N_{i_j} = \langle v_{i_j}, w_{i_j}, t_{i_j} \rangle \in N^C(v, w), v_{i_j} \in V\} \cup \{w\}$, and $E(v, w) = \{e_{i_j} = \langle v_{i_j}, w_{i_j}, g_{i_j}^E, s_{i_j}^E \rangle \,|\, i_j \in \mathbb{N}^*, \forall N_{i_j} = \langle v_{i_j}, w_{i_j}, t_{i_j} \rangle \in N^C(v, w), e_{i_j} = \langle v_{i_j}, w_{i_j}, g_{i_j}^E, s_{i_j}^E \rangle \in E\}$.

Definition 8. Chain of resource transmission refers to the whole courses of information exchange in the process of resource transmission. Resources exchange action means one resource transmission between two access requesting entities, denoted by $I = \langle s \rightarrow r, o, t\rangle$, where s, o, r and t represent resource sender or resource transmitter, resource receiver, resource itself and general tense as defined in the Def.2, respectively.

A chain of a resource transmission indicates an ordered set of the resources exchange actions, denoted by $O_i^C(s_i, r_i, o) = \{\langle I_{i,1}, I_{i,2}, \cdots, I_{i,k},\rangle \,|\, o \in O, I_{i,l} = \langle s_{i,l} \rightarrow r_{i,l}, o, t_{i,l}\rangle, s_{i,l} = r_{i,l+1}, s_{i,1} = s_i, r_{i,k} = r_i, s_i \in Q, r_i \in Q, s_{i,l} \in Q, r_{i,l} \in Q, 1 < l \leqslant k-1\}$, where s_i, r_i and o represent initiator of resource transmission, receiver of resource transmission and resource itself, respectively.

We define set of chains of resource transmission as $O^C = \{O_i^C(s, r_i, o) \,|\, i \in \mathbb{N}^*\}$, where initiator of resource transmission is represented by s, and receivers of resource transmission are denoted by $r_1, r_2, \cdots, r_i, \cdots$.

Definition 9. Scene sc refers to set of all permissions the information access entity needed to perform a function. It means general tense t, access point l, device d and network information ng the access entity q needed when q starts session s, where $t \in T$, $l \in L$, $d \in D$ and $ng \in NG$. Thus we denote sc as $sc = (t, l, d, ng)$. In this set, the valuing range of every elements in sc is $\{t^{SC}\}$, $\{l^{SC}\}$, $\{d^{SC}\}$ and $\{ng^{SC}\}$.

Definition 10. Scene constraints *Constraints-sc* represents that permissions are only obtainable through scene sc when the access entity initiate session s after initiating a session.

Definition 11. Resource access entity-scene assignment *qsc* refers to the process that resource access entity is assigned scene sc. We define the corresponding set as QSC.

Definition 12. Entity scene-permission assignment *qscp* refers to the process that *Constraints-qsc* is assigned permission p. We define the corresponding set as $QSCP$.

We present *Constraints-sc* in Fig. 2, where *enable*, *disable* and *active* are similar to the corresponding definition in [11]. S represents session set. *assign/deassign* represents relation of assignment and de-assignment. N_{active} means the active number. N_{max} denotes maximal active number. $active_{Q-total}$ represents the number of all scenes activated by resource access entity. $active_{P-total}$ represents the number of all activated scenes with a permission.

Constraints Type	Constraints		Description
SC-Enable Constraints	Q-SC Constraint		$(Q, S, T, L, D, NG,$ $\text{assgin}_Q/deassign_Q$ sc to $q)$
	SC-Enable		$(Q, S, T, L, D, NG,$ *enable/disable sc*$)$
	QSC-P Assignment		$(Q, S, T, L, D, NG,$ $assgin_P/deassign_P$ *p to sc*$)$
SC-Active Constraint	Number of Active SC	U	$(Q, S, T, L, D, NG, N_{active}, active_{Q\_total})$
		P	$(Q, S, T, L, D, NG, N_{active}, active_{P\_total})$
	Total Number of Active SC	U	$(Q, S, T, L, D, NG, N_{max}, active_{Q\_total})$
		P	$(Q, S, T, L, D, NG, N_{max}, active_{P\_total})$

Figure 2 Scene Constraints Description

3.3 Hierarchy Structure and its Inheritance Mechanism

Since scene comprehensively takes four factors into consideration, including general tense, access point, access device and network, it also has hierarchy structure.

Definition 13. General tense hierarchy structure refers to partial order over general tense set T, defined by $TH \subseteq T \times T$. For any t_i, $t_j \in T$, there is $(t_i, t_j) \in TH$ if and only if $t_i \geqslant t_j$. For $(t_i, t_j) \in TH$, we

define that t_i is higher than t_j, and denote $t_i \geqslant t_j$. If for $(t_i, t_j) \in TH$ there exits no t_k such that $t_i \geqslant t_k$ and $t_k \geqslant t_j$, we define that t_i is directly higher than t_j.

Definition 14. Access point hierarchy structure refers to partial order over access point set L, defined by $LH \subseteq L \times L$. For any l_i, $l_j \in L$, there is $(l_i, l_j) \in LH$ if and only if $l_i \geqslant l_j$. For $(l_i, l_j) \in LH$, we define that l_i is higher than l_j, and denote $l_i \geqslant l_j$. If for $(l_i, l_j) \in LH$ there exits no l_k such that $l_i \geqslant l_k$ and $l_k \geqslant l_j$, we define that l_i is directly higher than l_j.

Definition 15. General device hierarchy structure refers to partial order over general device set D, defined by $DH \subseteq D \times D$. For any d_i, $d_j \in D$, there is $(d_i, d_j) \in DH$ if and only if $d_i \geqslant d_j$. For $(d_i, d_j) \in DH$, we define that d_i is higher than d_j, and denote $d_i \geqslant d_j$. If for $(d_i, d_j) \in DH$ there exits no d_k such that $d_i \geqslant d_k$ and $d_k \geqslant d_j$, we define that d_i is directly higher than d_j.

Definition 16. Network hierarchy structure refers to partial order over network set NG, defined by $NGH \subseteq NG \times NG$. For any ng_i, $ng_j \in NG$, there is $(ng_i, ng_j) \in NGH$ if and only if $ng_i \geqslant ng_j$. For $(ng_i, ng_j) \in NGH$, we define that ng_i is higher than ng_j, and denote $ng_i \geqslant ng_j$. If for $(ng_i, ng_j) \in NGH$ there exits no ng_k such that $ng_i \geqslant ng_k$ and $ng_k \geqslant ng_j$, we define that ng_i is directly higher than ng_j.

Definition 17. Scene hierarchy structure refers to partial order over scene set SC, defined by $SCH \subseteq SC \times SC$. For any sc_i, $sc_j \in SC$, there is $(sc_i, sc_j) \in SCH$ if and only if $sc_i \geqslant sc_j$. For $(sc_i, sc_j) \in SCH$, we define that sc_i is higher than sc_j, and denote $sc_i \geqslant sc_j$. If for $(sc_i, sc_j) \in SCH$ there exits no sc_k such that $sc_i \geqslant sc_k$ and $sc_k \geqslant sc_j$, we define that sc_i is directly higher than sc_j.

3.4 Cyberspace-oriented Access Control Model

According to the definition of scene and formal definition in [11], we construct access control model as follows.

Definition 18. The components of Cyberspace-Oriented Access Control (CoAC) model are as follows:

1) Q, SC, P, S (ARE, scene, permission, session), $SC = (T, L, D, NG)$, where T, L, D, NG denotes general tense, access point, resource, access device, network as defined in 3.2.

2) $QSC \subseteq Q \times SC$ represents relation of assignment between many-to-many access requiring entity and scene.

3) $QSCP \subseteq QSC \times P$ denotes relation of assignment between many-to-many entity scene and permissions.

4) $entity : session \to Q$ represents the function mapping session s_j into access requiring entity.

5) $scene : S \to 2^{SC}$ denotes the function mapping session s_j into scene set. In this function, $scene(s_i) \subseteq \{sc \,|\, (\exists sc' \geqslant sc)\, [entity(s_i), a'] \in QSC\}$, and s_i has permission $\cup_{SC \in scene(s_i)} \{p \,|\, (\exists q'' \leqslant q)[(q'', p) \in SCP]\}$.

As shown in Fig. 3, the structure of CoAC model including various levels of scene, access point, general tense, access device and network.

Internet-based interactive graph limits access permissions to resource for access requiring entity through restricting transmission routes of resource on network. When access requiring entity q proposes access requirement of resource o to the resource sever, the sever will gets some information about q's network location simultaneously such as IP address, port number and etc. Then the sever set itself as the starting point and initiate "route discovery" to the ending point q. If the discovered route is included in resource o's network interactive graph, this access requirement, satisfying access control strategy, will be permitted. Otherwise, it will be refused. For example, a senior executive of a company is able to access to resource o upon access control strategy. However, when the senior executive travels extensively, because the link between the senior executive and his company isn't included in resource o's network interactive graph, the senior executive cannot access resource o.

Chain of resource transmission realize controlled secondary/mutiple distribution, information protection and trace to the source by limiting resource's forwarding path. When access requiring entity q proposes access requirement of resource o to the resource sever, the sever will judge whether q is in the chain of o's resource transmission or not. If q is included in the chain of o's resource transmission, the

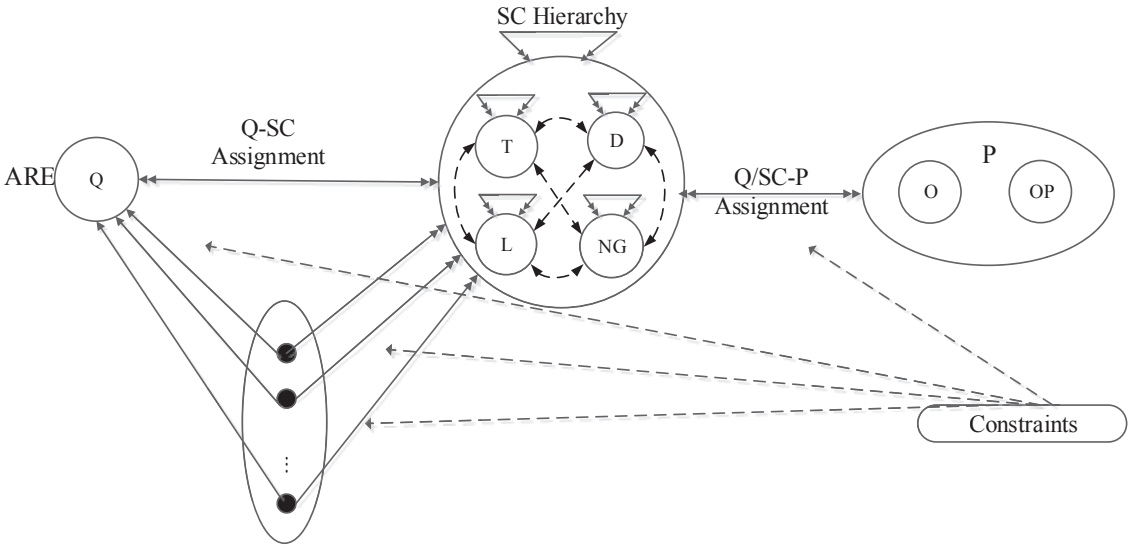

Figure 3 Access Model of CoAC

access requirement, satisfying access control strategy, will be permitted. Otherwise, it will be refused. Moreover, when q make a request for the operation permission of forwarding o to the resource sever, the sever will judge whether the receiver of o is in the chain of o's resource transmission or not. If yes, permit forwarding, otherwise refuse. In this sense, we can achieve controlled secondary/mutiple distribution of resource o. We also trace to the source of resource via chain of resource transmission. Because that resource is transmitted between different AREs lead to the fact that information of every ARE will be recorded in metadata of resource or metadata returned to sever, we can trace to the source of a resource by getting the metadata.

4 Administrative model

4.1 Administrative scene and administrative model

In our CoAC model, the administrative scene is used to realize security management of other scenes. So safety of administrative scene itself, related to overall security of CoAC model, is a core issue. We use AQ as set of AREs of administrative resource and define administrative scene by setting limited general tense, access point, access device and network.

Definition 19. Administration scene *adsc* is a particular scene that satisfy all properties of scene, but its general tense, access point, access device and network are limited. It is denoted by $adsc \in ADSC$, where $ADSC = \{adsc = (aq, limt, liml, limd, limng) \mid aq \in AQ, limt \in LIMT, liml \in LIML, limd \in LIMD, limng \in LIMNG\}$. In this set, AQ, $LIMT$, $LIML$, $LIMD$ and $LIMNG$ represent administrative resource access entity, limited general tense, limited access point, limited access device and limited network.

Administration scene can provide more securing service and manage trust for general scenes by introducing trusted platform module to controlled scene. We set ADP as set of administrative permission and present administrative model of in view of ABAC [11].

Definition 20. The components of administrative model are as follows:

1) Q: resource access entity. SC: general scene set defined in Def.9. $ADSC$: administrative scene set defined in Def.19. P: permission set. ADP: administrative permission set. S: session set.

2) $SCSC \subseteq SC \times SC$ represents many-to-many scene-scene assignment relation.

3) $QSC \subseteq Q \times SC$ represents many-to-many resource access entity-scene assignment relation.

4) $QADSC \subseteq Q \times ADSC$ denotes many-to-many user-administrative scene assignment relation.

5) $ADSCP \subseteq ADSC \times P$ denotes many-to-many administrative scene-permission assignment relation.

6) $SCH \subseteq SC \times SC$ refers to partial order over scene set SC, denoted by $\geqslant$.

7) $ADSCH \subseteq ADSC \times ADSC$ represents partial order over administrative scene set $ADSC$, denoted as $\geqslant$.

8) *Constraints* denotes constraint conditions.

9) $entity : S \rightarrow Q$ represents the function mapping session s_i into access requiring entity $entity(s_i)$. (Session remains unchanged in its lifecycle).

10) $scene : S \rightarrow 2^{SC \cup ADSC}$ denotes the function mapping session s_i into scene set. In this function, $scene(s_i) \subseteq \{sc \,|\, (\exists sc' \geqslant sc)\,[(entity(s_i), a') \in QSC \cup QADSC]\}$, and s_i has permission $\cup_{SC \in scene(s_i)}\{p \,|\, (\exists q'' \leqslant q)[(q'', p) \in SCP \cup ADSCP]\}$.

Fig. 4 presents structure of scene administrative model, including various levels of general scene, administrative scene, general tense, access device and network. The figure shows that scene administrative model use administrative scene set $ADSC$ to manage resource access entity-scene assignment, scene-permission assignment and scene state. Scene administrative model controls resource access entity-scene, scene-permission, resource access entity-administrative scene assignment, administrative scene-administrative permission assignment through *Constraints*. Since administrative scene manage general scene, the security of management is able to be guaranteed.

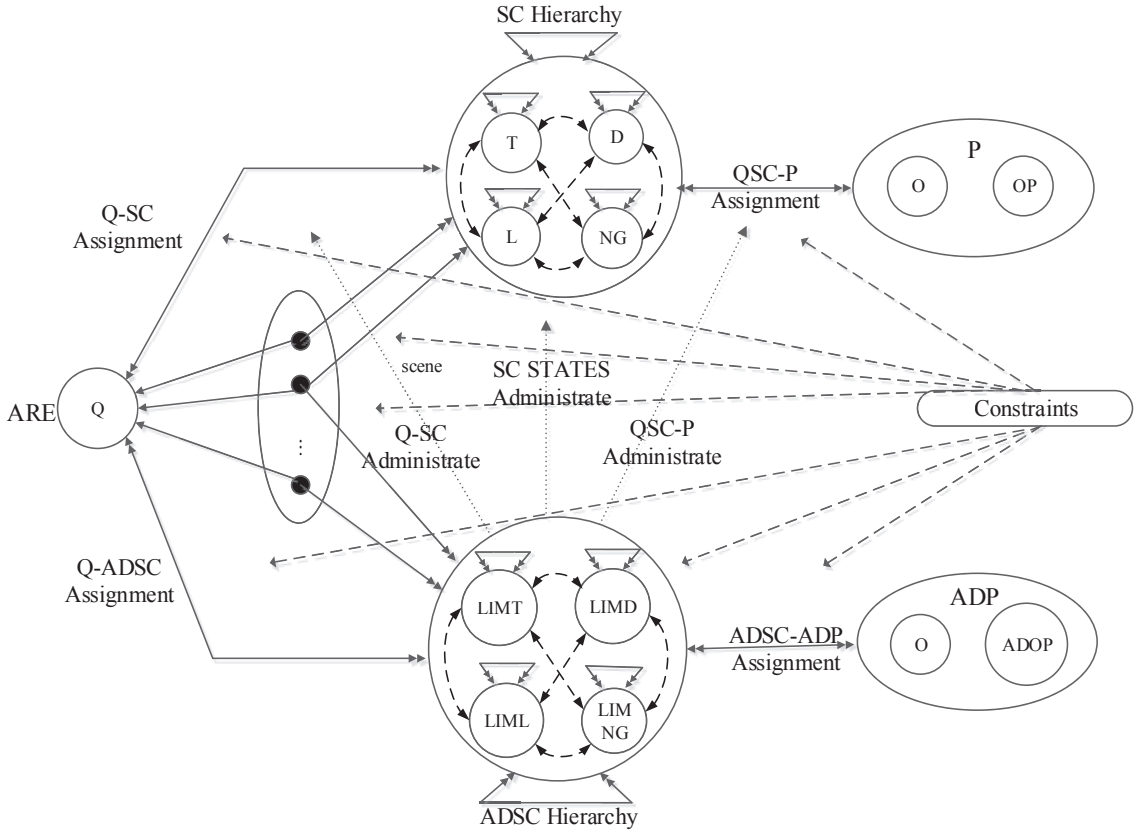

Figure 4 Administrative Model of CoAC

4.2 Functions of administrative model

We have proposed control relationship among resource access entity-scene assignment, resource access entity-scene revocation, scene-permission assignment and scene-permission revocation in administrative model. Based on it, we define control relationship among resource access entity-administrative scene control relationship, administrative scene-administrative permission and function related to scene state administration in CoAC.

Definition 21. We denote prerequisite as Boolean-expression that operate with x and $\overline{x}$ through operator $\vee$ and $\wedge$.

1) $x \in SC$ represents general scene. For resource ARE q, if x is true, we have $(\exists x' \geqslant x)((q,x') \in QADSC)$. If $\overline{x}$ is true, we have $(\forall x' \geqslant x)((q,x') \notin QA \vee (q,x') \notin QADSC)$. For given general scene set SC, set CSC as all prerequisite conditions possibly gain by using SC.

2) $x \in ADSC$ represents administrative scene. For resource ARE q, if x is true, we have $(\exists x' \geqslant x)((q,x') \in QADSC \vee (q,x') \notin QA)$. If $\overline{x}$ is true, we have $(\forall x' \geqslant x)((q,x') \notin QADSC \vee (q,x') \notin QA)$. For given administrative scene set $ADSC$, set $CADSC$ as all prerequisites administrative scene possibly gain by using $ADSC$.

Definition 22. Administration model uses following relations to assign resource access entity-administrative scene and revoke resource access entity-administrative scene that has already assigned.

$$can.assignsq \subseteq ADA' \times CADSC \times 2^{ADSC},$$

$$can.revokesq \subseteq ADA' \times 2^{\{ADSC \setminus \{superadsc\}\}}.$$

In the above relations, $ADA' = ADA\{a \mid \nexists a' \in ADA, a' < a\}$. *superadsc* represents system setting super administrative scene, located on the top of administrative scene hierarchy structure, and cannot be revoked. If Z denotes set of administrative scene could be allocated, $can.assignsq(x,y,Z)$ represents that administrative scene x (or $\forall x' > x$) can assign administrative scene $z \in Z$ to users satisfying prerequisite y. If Z denotes revoked administrative behavior set, $can.revokesq(x,Z)$ represents that administrative scene x (or $\forall x' > x$) can revoke administrative scene set Z assigned to user.

Definition 23. Administration model use following relations to assign administrative scene-administrative permission and revoke administrative scene-administrative permission that has already assigned.

$$can.assignsq \subseteq ADA' \times CADSC \times 2^{ADP},$$

$$can.revokesq \subseteq ADA' \times 2^{ADP}.$$

In the above relations, $ADA' = ADA\{a \mid \nexists a' \in ADA, a' < a\}$. $ADP = \{entityad, scenead, permissionad, qscad, scpad, scsad\}$ represents administrative permission set, where *entityad*, *scenead*, *permissionad*, *qscad*, *scpad*, *scsad* denotes resource access entity administration, scene administration, permission administration, user-scene assignment administration, scene-permission assignment administration and scene state administration, respectively. If Z denotes set of administrative permission could be allocated, $can.assignsq(x,y,Z)$ represents that administrative scene x (or $\forall x' > x$) can assign administrative permission $z \in Z$ to administrative scene satisfying prerequisite y. If Z denotes revoked administrative behavior set, $can.revokesq(x,Z)$ represents that administrative scene x (or $\forall x' > x$) can revoke administrative scene set Z assigned to user.

We apply Z-notation to formalized definition of add, revise and delete scene in scene state administration. *NAME* that denotes abstract data type can represent component of scene model such as scene, resource access entity, general tense, access point, access device, network, session, permission and etc. *USERS*, *QUERYS*, *DEVICES*, *NETGRAPHS*, *SESSIONS*, *OPS* and *OBS* means USER set, ARE set, access device set, network set, session set, operation set and object set, respectively.

$AddScene(scene : NAME)\triangleleft$

$scene \notin SCENES$

$if\ scene.query \notin QUERYS\ then\ QUERYS' = QUERYS \cup \{role\}$

$if\ scene.temporalstate \notin TSTAES\ then\ TSTATES' = TSTATES \cup \{temporalstate\}$

$if\ scene.locate \notin LOCATES\ then\ LOCATES' = LOCATES \cup \{locate\}$

$if\ scene.device \notin DEVICES\ then\ DEVICES' = DEVICES \cup \{device\}$

$if\ scene.netgraph \notin NETGRAPHS\ then\ NETGRAPHS^{'} = NETGRAPHS \cup \{netgraph\}$

$SCENES' = SCENES \cup \{scene\}$

$QSC' = QSC \cup \{scene \to \varnothing\}$

$SCP' = SCP \cup \{scene \to \varnothing\}\triangleright$

$ModifyScene(scene, query, temporalstate, locate, device, netgraph : NAME)\triangleleft$

$scene \in SCENES$

$if\ scene.query \notin QUERYS\ then\ QUERYS' = QUERYS \cup \{query\}$

$if\ scene.temporalstate \notin TSTAES\ then\ TSTATES' = TSTATES \cup \{temporalstate\}$

$if\ scene.locate \notin LOCATES\ then\ LOCATES' = LOCATES \cup \{locate\}$

$if\ scene.device \notin DEVICES\ then\ DEVICES' = DEVICES \cup \{device\}$

$if\ scene.netgraph \notin NETGRAPHS\ then\ NETGRAPHS^{'} = NETGRAPHS \cup \{netgraph\}$

$[\forall s \in SESSIONS\ scene \in SCENE\ scene(s) \Rightarrow DeleteSession(s)]$

$scene' = (temporalstate, locate, device, netgraph)$

$SCENES' = SCENES \setminus \{scent\} \cup \{scene'\}$

$QSC' = QSC \setminus \{\forall q \in QUERYS \cdot scene \mapsto q\} \cup \{scene' \mapsto \varnothing\}$

$SCP' = SCP \setminus \{\forall op \in OPS. \forall ob \in OBS \cdot scene \mapsto (op, ob)\} \cup \{scene' \mapsto \varnothing\}\triangleright$

$DeleteScene(scene : NAME)\triangleleft$

$scene \in SCENES$

$[\forall s \in SESSIONS \cdot scene \in session.scene(s) \Rightarrow DeleteSession(s)]$

$[\forall sc \in SCENES \setminus \{scene\} \cdot sc.q \neq scene.q \Rightarrow QUREYS' = QUERYS \setminus \{scene.q\}]$

$[\forall sc \in SCENES \setminus \{scene\} \cdot sc.temporalstate \neq scene.temporalstate \Rightarrow TSTATES' = TSTATES \setminus \{scene.temporalstate\}]$

$[\forall sc \in SCENES \setminus \{scene\} \cdot sc.locate \neq scene.locate \Rightarrow LOCATES' = LOCATES \setminus \{scene.locate\}]$

$[\forall sc \in SCENES \setminus \{scene\} \cdot sc.device \neq scene.device \Rightarrow DEVICES' = DEVICES \setminus \{scene.device\}]$

$[\forall sc \in SCENES \setminus \{scene\} \cdot sc.netgraph \neq scene.netgraph \Rightarrow NETGRAPHS' = NETGRAPHS \setminus \{scene.netgraph\}]$

$QSC' = QSC \setminus \{\forall q \in QUERYS \cdot scene \mapsto q\}$

$SCP' = SCP \setminus \{\forall op \in OPS. \forall ob \in OBS \cdot scene \mapsto (op, ob)\}$

$SCENES' = SCENES \setminus \{scene\}\triangleright$

We can identify ARE's user identity, access agent, role information, access point, access device, network and etc, and use functions in the following table to manage ARE-scene, scene-permission as well as the state of scene.

Table 1 Administrative Functions

Function Name	Description
verifyid	If the user's identify id is valid, return True, otherwise return False. $verifyid(userid, certification : NAME.outresult : BOOLEAN)\lhd$ $result = (userid \in U) \wedge (isvalid(certification))\rhd$
isnable	If the scene is valid, return True, otherwise return False. $isnable(scene : NAME.outresult : BOOLEAN)\lhd$ $scene \in SCENE$ $result = scene \in ENABLE^*\rhd$
verifyt	If the general temporal state of request $reqt$ satisfies the general temporal state $verifyt$, return True, otherwise return false. $verifyt(reqt : NAME.outresult : BOOLEAN)\lhd$ $reqt \in TSTATES$ $result = (\exists t_1, t_2 \in validt \cdot t_1 < reqt < t_2) \wedge (q \in Q.t \in TSTATES.l \in LSTATES.$ $d \in DSTATES.ng \in NGSTAGES.p \in PERMISSIONS \cdot (q, sc = (reqt, l, d, ng) \in QSC)$ $\wedge (sc = (reqt, l, d, ng), p) \in SCP)\rhd$
verifyl	If the access point of request state $reql$ satisfies the access point state $verifyl$, return True, otherwise return false. $verifyl(reql : NAME.outresult : BOOLEAN)\lhd$ $reql \in LSTATES$ $result = (\exists l_1, l_2 \in validt \cdot l_1 < reql < l_2) \wedge (q \in QUREYS.l \in LSTATES.t \in TSTATES.$ $d \in DSTATES.ng \in NGSTAGES.p \in PERMISSIONS \cdot (q, sc = (t, reql, d, ng) \in QSC)$ $\wedge (sc = (t, reql, d, ng), p) \in SCP)\rhd$
verifyd	If the device state of request $reqd$ satisfies the device state $verifyd$, return True, otherwise return false. $verifyd(reqd : NAME.outresult : BOOLEAN)\lhd$ $reqd \in DSTATES$ $result = (\exists d_1, d_2 \in validd \cdot d_1 < reqd < d_2) \wedge (q \in QUERYS.d \in DSTATES.l \in LSTATES.$ $t \in TSTATES.ng \in NGSTAGES.p \in PERMISSIONS \cdot (q, sc = (t, l, reqd, ng) \in QSC)$ $\wedge (sc = (t, l, reqd, ng), p) \in SCP)\rhd$
verifyng	If the network state of request $reqt$ satisfies the network state $verifyng$, return True, otherwise return false. $verifyng(reqng : NAME.outresult : BOOLEAN)\lhd$ $reqng \in NGSTATES$ $result = (\exists ng_1, ng_2 \in validt \cdot ng_1 < reqng < ng_2) \wedge (q \in QUREYS.ng \in NGSTAGES.$ $t \in TSTATES.l \in LSTATES.d \in DSTATES.p \in PERMISSIONS$ $\cdot (q, sc = (t, l, d, reqng) \in QSC) \wedge (sc = (t, l, d, reqng), p) \in SCP)\rhd$
n.activescbyu	Return the active scene number of the user. $n.activescbyu(user : NAME.outresult : N)\lhd$ $user \in USERS$ $result = N_{active\_U\_total}(user)\rhd$
maxn.activescbyp	Return the maximal active scene number of the user. $maxn.activescbyu(user : NAME.outresult : N)\lhd$ $user \in USERS$ $result = N_{max\_U\_total}(user)\rhd$
n.activescbyp	Return the scene number of the active permission. $maxn.activescbyu(permission : NAME.outresult : N)\lhd$ $permission \in PERMISSIONS$ $result = N_{active\_P\_total}(user)\rhd$
maxn.activescbyp	Return the scene number of the active permission. $maxn.activescbyu(permission : NAME.outresult : N)\lhd$ $permission \in PERMISSIONS$ $result = N_{max\_P\_total}(user)\rhd$

5 Descriptions of Some Typical Models by Using CoAC

In this section, we present how to use our proposed CoAC model to describe some typical access control models. Generally, since our CoAC contains a set of vital factors, they can be easily adjusted to cover most of existing models, such as DAC, MAC, TRBAC/RBAC and ABAC.

5.1 Discretionary Access Control (DAC)

According to the Def1 and Def4, let $A = R = G^O = S^O = \varnothing$. Then the DAC model is equivalent to $\langle Q, O, OP\rangle$, that is $DAC \cong \langle Q, O, OP\rangle$, where $Q = \{q = \langle u, \cdot, \cdot\rangle\}$ denotes the subject in DAC model,

$O = \{o = \langle c^O, \cdot, \cdot \rangle\}$ denotes the object in DAC model, OP denotes the operational type.

5.2 Mandatory Access Control (MAC)

According to the Def1 and Def4, let $A = S^O = \varnothing$. Then the MAC model is equivalent to $\langle Q, O, OP \rangle$, that is $MAC \cong \langle Q, O, OP \rangle$, where $Q = \{q = \langle u, \cdot, r \rangle\}$ denotes the subject in DAC model, $O = \{o = \langle c^O, \cdot, s^O \rangle\}$ denotes the object in MAC model, OP denotes the operational type.

5.3 Temporal Role-based Access Control (TRBAC) / Role-based Access Control (RBAC)

According to the Def1, Def2 and Def4, let$G^O = S^O = \varnothing$. Then the TRBAC model is equivalent to $\langle Q, O, T, OP \rangle$, that is $TRBAC \cong \langle Q, O, T, OP \rangle$, where $Q = \{q = \langle u, \cdot, r \rangle\}$ and u, r denote the user and role in TRBAC model respectively, $O = \{o = \langle c^O, \cdot, \cdot \rangle\}$ denotes the resources in the TRBAC model, OP denotes the operational type.

As above mentioned, if we let $T = \varnothing$, then the TRBAC model converts to RBAC model. In another words, RBAC model is equivalent to $\langle Q, O, OP \rangle$, that is $RBAC \cong \langle Q, O, OP \rangle$.

5.4 Action-based Access Control (ABAC)

According to Def1-6, let $S^D = G^O = S^O = \varnothing$. Then the ABAC model is equivalent to $\langle Q, O, A, OP \rangle$, that is $ABAC \cong \langle Q, O, A, OP \rangle$ where $Q = \{q = \langle u, a, r \rangle\}$ and u, a, r denote the user, agent and role in the ABAC model, $O =$ denotes the resources in the ABAC model, $A = \langle R, L, D, NG, T \rangle$ and R, L, $D = \{d = \langle g^D, \cdot, t \rangle\}$, NG, T denote the role, device, network, and general temporal respectively.

6 Conclusions

To solve the new problems brought by the novel information service patterns and dissemination modes, i.e., the ownership of data is separated from the administration of them, secondary/mutiple information distribution etc., we proposed a novel Cyberspace-oriented Access Control model, named CoAC. It considers some vital factors in the access control models, including the access requesting entity, general tense, access point, resource, device, networks, internet-based interactive graph and chain of resource transmission. Thus, CoAC can cover most of existing access control models such as DAC, RBAC, MAC, ABAC, etc. CoAC is also adapted to most new service patterns such as distributed computing, mobile computing and cloud computing, etc., by appropriately adjusting the aforementioned factors, which makes people easily access and manage resources anytime, anywhere with any devices and via any internet. Meanwhile, we introduced the definition of administrative scene and administrative model of CoAC, which can achieve fine-grained, multi-level and flexible information sharing and data access control. The CoAC is flexible and scalable which can be further refined and expanded to serve the development of the new information service patterns and dissemination modes in the future.

References

1 S. Berg, C. Lane, and M. Whittaker, *Glossary of Computer Security Terms.* National Computer Security Center, 1989.

2 D. Bell and L. LaPadula, "Secure computer systems: Mathematical foundations and model."

3 D. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-based access control.* Artech House, 2003.

4 R. K. Thomas and R. S. Sandhu, "Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management," *DBSec*, vol. 113, pp. 166–181, 1997.

5 E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti, "drbac: distributed role-based access control for dynamic coalition environments," in *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on.* IEEE, 2002, pp. 411–420.

6 S. Liu and H. Huang, "Role-based access control for distributed cooperation environment," in *Computational Intelligence and Security, 2009. CIS'09. International Conference on*, vol. 2. IEEE, 2009, pp. 455–459.

7 E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca, "Geo-rbac: a spatially aware rbac," in *Proceedings of the tenth ACM symposium on Access control models and technologies.* ACM, 2005, pp. 29–37.

8 C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Supporting location-based conditions in access control policies," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. ACM, 2006, pp. 212–222.

9 E. Yuan and J. Tong, "Attributed based access control (abac) for web services," in *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE, 2005.

10 M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 99–112.

11 F. Li, W. Wang, J. Ma, and X. Liang, "Action-based access control model and administration of actions," *Acta Electronica Sinica*, vol. 36, no. 10, pp. 1881–1890, 2008.

12 M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 121–130.

13 X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: secure multi-owner data sharing for dynamic groups in the cloud," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 1182–1191, 2013.

14 W. Stallings, *Network and internetwork security: principles and practice*. Prentice Hall Englewood Cliffs, 1995, vol. 1.

15 R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

16 R. Sandhu, V. Bhamidipati, and Q. Munawer, "The arbac97 model for role-based administration of roles," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 1, pp. 105–135, 1999.

17 R. Sandhu and Q. Munawer, "The arbac99 model for administration of roles," in *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual*. IEEE, 1999, pp. 229–238.

18 S. Oh, R. Sandhu, and X. Zhang, "An effective role administration model using organization structure," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 2, pp. 113–137, 2006.

19 E. Bertino, E. Ferrari, and V. Atluri, "The specification and enforcement of authorization constraints in workflow management systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 1, pp. 65–104, 1999.

20 G. Coulouris, J. Dollimore, and M. Roberts, "Role and task-based access control in the perdis groupware platform," in *Proceedings of the third ACM workshop on Role-based access control*. ACM, 1998, pp. 115–121.

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

22 X.-M. Wang and Z.-T. Zhao, "Role-based access control model of temporal object." *Dianzi Xuebao(Acta Electronica Sinica)*, vol. 33, no. 9, pp. 1634–1638, 2005.

23 C. Xu, Q. Wang, W. Zhang, and Y. Ding, "Temporal access control based on multiple subjects," in *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, vol. 2. IEEE, 2009, pp. 438–441.

24 L. Xiao-feng, "4, feng deng-guo1, 2, chen zhao-wu3, 4, fang zi-he4 (1. state key laboratory of information security, institute of software of chinese academy of sciences, beijing 100080, china; 2. state key laboratory of information security, graduate school of chinese academy of sciences, beijing 100039, china; 3. beijing zhongdun security technology development co., beijing 100044, china 4. the first research institute of ministry of public security of prc, beijing 100044, china); model for attribute based access control [j]," *Journal on Communications*, vol. 4, 2008.

25 X.-m. WANG, H. FU, and L.-c. ZHANG, "Research progress on attribute-based access control [j]," *Acta Electronica Sinica*, vol. 7, p. 033, 2010.

26 T. May, "Time-release crypto," 1993.

27 R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," 1996.

28 J. Cathalo, B. Libert, and J.-J. Quisquater, "Efficient and non-interactive timed-release encryption," in *Information and Communications Security*. Springer, 2005, pp. 291–303.

29 K. G. Paterson and E. A. Quaglia, "Time-specific encryption," in *Security and Cryptography for Networks*. Springer, 2010, pp. 1–16.

30 L. Zhou, V. Varadharajan, and M. Hitchens, "Enforcing role-based access control for secure data storage in the cloud," *The Computer Journal*, p. bxr080, 2011.

31 G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 735–737.

32 C. Hong, M. Zhang, and D. Feng, "Ab-accs: A cryptographic access control scheme for cloud storage," *Journal of Computer Research and Development*, vol. 47, pp. 259–265, 2010.

33 Y. Cheng, J. Ren, Z. Wang, S. Mei, and J. Zhou, "Re-encryption optimization in cp-abe based cryptographic cloud storage," in *Cloud and Green Computing (CGC), 2012 Second International Conference on*. IEEE, 2012, pp. 173–179.

34 J. Ye, W. Zhang, S.-l. Wu, Y.-y. Gao, and J.-t. Qiu, "Attribute-based fine-grained access control with user revocation," in *Information and Communication Technology*. Springer, 2014, pp. 586–595.

35 P.-P. Wang, D.-G. Feng, and L.-W. Zhang, "Cp-abe scheme supporting fully fine-grained attribute revocation," *Ruanjian Xuebao/Journal of Software*, vol. 23, no. 10, pp. 2805–2816, 2012.

36 M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography*. Springer, 2007, pp. 515–534.