Hindawi
Security and Communication Networks
Volume 2022, Article ID 6181995, 15 pages
https://doi.org/10.1155/2022/6181995

Research Article

Research on Access Control Scheme of System Wide Information Management Based on Attribute Association

Lizhe Zhang,[1,2] **Zhenghang You**,[2] **Kenian Wang**,[1,2] **and Zihan Cui**[2]

[1]*Key Laboratory of Civil Aircraft Airworthiness Technology, Civil Aviation University of China, Tianjin 300300, China*
[2]*School of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China*

Correspondence should be addressed to Lizhe Zhang; lzzhang@cauc.edu.cn

Received 4 March 2022; Accepted 6 May 2022; Published 30 May 2022

Academic Editor: B. Wang

System wide information management (SWIM) involves civil aviation system control, intelligence, alarm, traffic, and other data. These data are transmitted in various forms, making SWIM system vulnerable to sensitive information leakage, data tampering, denial, and other security threats. In this article, an attribute-based air traffic management (ATM) information access control scheme is proposed to solve the security threat of SWIM. An improved extensible access control markup language (XACML) authorization model is established, combining linear secret sharing scheme (LSSS) matrix structure and monotone span program (MSP); an attribute association algorithm is designed to establish the attribute association relationship between services and users. Experimental results show that the attribute association algorithm improves the time complexity, but the algorithm can support richer policy representation capability, and the proposed ATM information access control scheme is more efficient and can effectively reduce the space cost. This scheme can achieve more fine-grained and flexible access control.

1. Introduction

System wide information management (SWIM) is a distributed system based on service-oriented architecture. It enables data conversion, service integration, and information transformation for civil aviation systems, which facilitates civil aviation to establish a unified and efficient data interaction platform [1]. With the development of SWIM, we realize that the main goal of SWIM is not only to enable seamless integration between geographically distributed and heterogeneous systems in the air transport sector but also to enable seamless information sharing among multiple stakeholders in the ATM domain [2]. In terms of the advantage of SWIM, it not only effectively prevents the "information island" problem caused by heterogeneous systems and reduces the enormous overhead caused by interconnection and maintenance but also enhances the collaborative decision-making and situational awareness capabilities of civil aviation units [3].

SWIM can facilitate the development of civil aviation. However, viruses, network attacks, and other means are easy to find system security vulnerabilities and then threaten the system operation [4]. Increasing data are transmitted and exchanged in the SWIM network, such that the security threat is becoming considerably prominent [5]. For SWIM, the most important threats are data leakage and privacy protection. Considering that SWIM service data involve sensitive information in the airspace of member states and the commercial privacy of airlines, all parties' concern about information security has become a major obstacle to the development of SWIM. Preventing data leakage and protecting user privacy are the most important issues. The main reason for these problems is the occurrence of illegal access and unauthorized access. To alleviate the impact of the above problems, access control has become one of the key technologies of SWIM security services [6].

2. Related Work

In the aspect of information access control, many scholars have carried out a series of research. In view of the characteristics of big data resources and the problems existing in

the centralized access control mechanism, Liu et al. [7] proposed a blockchain-based big data access control mechanism based on the ABAC model, which adopts the access control method based on smart contract to achieve transparent, dynamic, and automatic access control over big data resources. Du et al. [8] proposed a hierarchical block chain-based distributed architecture for the Internet of things, which is based on ABAC model and adopts smart contract to realize dynamic and automatic access control for the devices of the Internet of things within and across domains based on attributes. Zhang [9] designed an attribute matching method based on Bloom filter on the original access control scheme, which greatly improved the access control efficiency. Wu [10] proposed a risk-based XACML access control model for fog computing. Based on this model, risk assessment method and privacy policy adaptive method were proposed to ensure the security of private data in fog computing. Han et al. [11] proposed an XACML policy query method based on attribute and or matrix and type analysis to reduce the number of rule matching during the implementation of policy evaluation. Wang et al. [12] improved the original XACML-based access control policy conflict detection method and designed a new one-time resolution algorithm for policy conflict. Luo et al. [13] proposed a metamodel-based access control policy description language (PML) and its implementation mechanism (PML-EM), which reduced the cost for users to write policies and for cloud service providers to develop access control mechanisms. Khaled and Cheng [14] studied the access control problem in the integrated distributed Internet of things environment and proposed an adaptive XACML scheme, which extended the typical XACML by integrating the access code generation and verification scheme in the heterogeneous distributed Internet of things environment.

In terms of security, although the security architecture and technical specifications in SWIM are not perfect, information security has already become a hot issue in the research of various countries. Boeing company of the United States pointed out that the security measures of the current SWIM system are not perfect, especially the protection measures for important private data, and proposed an information protection method based on SAML [15]. FAA and Embry-Red Aeronautic University identified security vulnerabilities related to data exchange between AAtS and NAS information service in SWIM, proposed security threats faced by AAtS, and established appropriate test methods [16]. Chris W. Johnson of the University of Glasgow proposed the potential security threats faced by NextGen and SESAR based on his analysis of them, aiming to promote their cooperation and provide mutual assistance in case of attack [17]. In order to ensure the safe operation of the new ATM system, EUROCONTROL developed a complete set of methods to support the integration of security into the design at the beginning of the development process [18].

Attribute is the core concept of ABAC. Attributes in ABAC can be described by a four-tuple (S, O, P, and E). S represents the subject attribute, that is all entities that the subject initiates access requests have attributes; O represents the object attribute, that is the resources that can be accessed in the system have attributes; P represents the permission attribute, that is all kinds of operations on object resources; E is the environment attribute, that is the environment information when access control occurs; the attributes of the subject and the object are used as the basic decision elements and used the attributes set of the requester to decide whether to grant access to them, which can well isolate policy management and access judgment, and effectively solve the problem of fine-grained access control in dynamic large-scale environment. According to the characteristics of SWIM and the actual business process of civil aviation in China, this article proposes an attribute-based access control (ABAC) scheme.

3. Overview of SWIM Services

SWIM services are mainly divided into core and application services. The core services are the basis of information security sharing within the system and are invisible. They mainly include interface management, security services, information transmission, and enterprise service management. The application services that expose the interface to other systems in accordance with the technical standards specified in SWIM are visible and discoverable. They are used to realize information interaction among systems [19]. SWIM data mainly exist in various application services, such that the services studied in this chapter belong to application services.

Given that ATM data are the core of SWIM service information and that service information belongs to the object resources in ABAC, they are an important part. This chapter mainly deals with the current ATM data as well as the composition, naming, and routing of service information in SWIM.

3.1. Composition of Service Information. As an intermediate layer between the underlying data and high-level applications, SWIM defines a unified data model and service standard for the entire ATM system, avoiding the heterogeneity of various types of information that were originally used only in specific domains. SWIM mainly divides data types in accordance with information usage and performs corresponding standardization work. The relationship between ATM data and SWIM service information is shown in Figure 1.

ATM system is a large socio-technical system, which adds a dimension to the cybersecurity complexity [20]. ATM is divided into three parts, namely airspace management, air traffic flow management, and air traffic service, which involve various types of data, such as regulation, intelligence, alarm, and traffic. In the current civil aviation network, these data are transmitted in various forms, including the Aeronautical Fixed Telecommunication Network, controller-pilot data link communications, and ATS Inter-device Data Communication. [19].

SWIM mainly developed the Flight Information Exchange Model, Weather Information Exchange Model, and Aeronautical Information Exchange Model. [21] These data

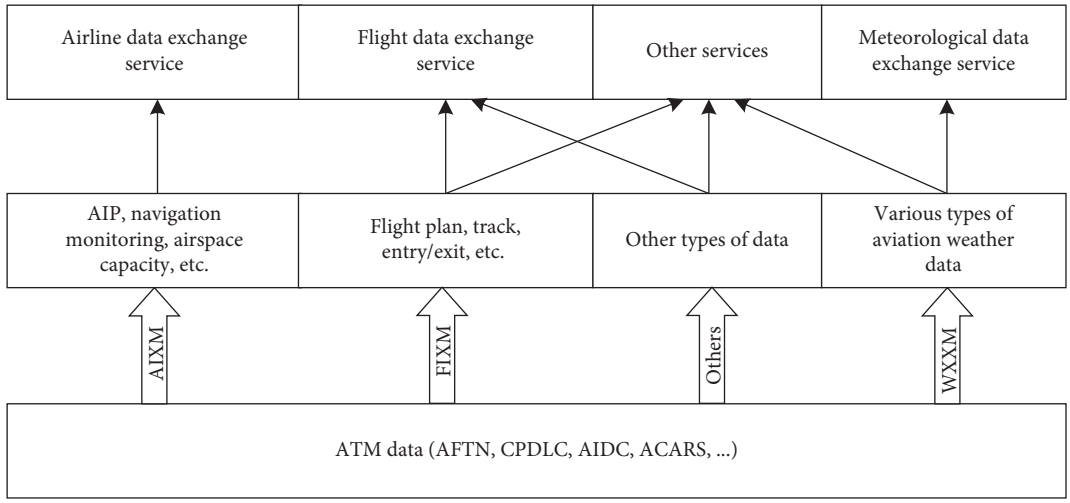


Figure 1: ATM data and SWIM service relationship diagram.

models are defined as UML and XML schemas. The raw data are converted to XML format based on the schema, and then exchanged between the application and the SWIM service. Moreover, metadata is necessary to obtain relevant information, such as provenance, type, time stamp, and quality. [22] ATM data are classified in accordance with the above models and formatted separately to conform to the standardized format requirements in SWIM. Given that the converted data are no longer heterogeneous, the data among different domains can be arbitrarily combined. The service definition only needs to focus on the data usage without having to pay attention to the source. SWIM services are in the definition stage, and ICAO divides services into four main categories: flight data exchange services, aeronautical data exchange services, weather data exchange services, and other services. The various types of services can be further divided into flight plans, control information, and alarm information in accordance with the specific use of the data provided.

3.2. Naming of Service Information. Attributes are the bases for naming SWIM service information and the core elements of ABAC. Therefore, this section first defines the attributes in SWIM. On the basis of the organizational structure of China's Air Traffic Control Bureau, airports, and airlines, as well as the current status of China's ATM network, the three attributes of users, resources, and environment in the civil aviation field are defined and assigned. The definition of the attributes is shown in Table 1.

Each of the users, services, and environments can be elucidated by combining the various attributes in Table 1. In addition, the attributes contained in the civil aviation field are relatively complicated, and some special attributes are not included in the table, such as Globally Unique Flight Identifier and flight number in flight information and airport four-character code in airport information.

As the infrastructure of the new generation of ATM, SWIM connects various aviation systems worldwide, which contain much aviation data. The amount of data is still growing rapidly with the further development of the aviation industry. To deal with the above problems, the typical hierarchical naming method uniform resource locator (URL) is selected. It has high aggregation characteristics to prevent the impact of massive data on the operation of the system. Owing to the wide variety of services in SWIM, only the naming schemes for several typical services are listed in Table 2.

As shown in Table 2, the elements in the naming are derived from the defined SWIM attributes, which enhance the semantics of naming and make it easy for users to find the information they need. SWIM uses the longest prefix matching method to request service information, such that information naming should be defined in order in accordance with the degree of differentiation of attributes. The head of the name is from the type of SWIM information to indicate the purpose. Afterward, the flight information requires the flight number to indicate the aircraft number, the aeronautical information requires the organization name to fix the source of the information, and the weather information needs to fix the area indicated by the information. Different types of information have distinct naming methods due to their differences in use and content.

From the above naming rules, hierarchical naming can achieve information aggregation at different levels, and the requester can directly match multiple pieces of information through a prefix. For example, Air China's executives can subscribe to information at the agency level, while North China Bureau's Flight Services Reporting Room can only subscribe to services at the regional level. This approach is also the basis for making access control highly flexible.

3.3. Routing of Service Information. To improve the efficiency of network transmission, SWIM transforms the traditional store-and-forward mode into a cache-forward mode [23]. While the service node is routing, it stores a copy of the information on some nodes in the transmission path. When the neighboring user needs this information again, it can be

Table 1: Attribute definition.

Attribute	Attribute classification	Attribute value
Region	Users, resources	Air traffic management area
City	Users, resources	City name
Organization	Users, resources	Organization name
Department	Users	Department name
Position	Users	Position name
Level	Users	User level
Class	Resources	Information class
Urgency	Resources	Urgency fact
Time	Environment	Current time
Network status	Environment	Current network status

Table 2: SWIM service information naming.

Type of information	Naming rules
Flight plan	/information type/organization/departure city/flight number/date/remarks
Track information	/information type/GUFI/control sector/date/remarks
Aeronautical information compilation	/information type/target (route/airport)/validity/remarks
Weather information	/information type/region/city/emergency/date/remarks

obtained in the cache of the short-distance node, such that each information transmission has the relatively shortest routing distance. The routing mechanism of SWIM is shown in Figure 2.

A three-part structure exists in the SWIM service node: Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS) [24]. The FIB is a base of forwarding information, the PIT stores the information in the packet or the interface set, and the CS keeps copies of the passed information.

When a visitor wants some information, it will send an Interest packet to the service node. The node looks for the information that matches the name in the CS. If it is found, the incoming interface is returned, and the information packet is discarded. If not, the node will look up the PIT to ensure that there is the same Interest packet that is forwarded from other nodes. If found, the entry is added to the PIT; if not, the FIB is searched. In the FIB, the node looks for the longest matching prefix and routes to determine the path of forwarding packets. When it is found, the node creates the information entry item in the PIT and forwards the packet. Otherwise, the information packet will be discarded.

When a Data packet arrives at the content router, the longest prefix is compared with the CS item. If the same cache data exist, it will be discarded or compared with the PIT. If there is a matching entry in the PIT, the Data packet is forwarded to the corresponding interface, and the packet is buffered in the CS table. If there is no matching entry in the PIT, the packet is discarded. The routing path of Data packets is just opposite to that of Interest packets.

3.4. SWIM Security Threat. SWIM has an open network environment and faces many security risks. This can lead to problems such as theft of trade secrets, maliciously tampering of aviation information, and disclosure of private data. These problems destroy the security of aviation data and affect the normal operation of civil aviation system. SWIM is subject to the following threats in terms of privacy protection and data security:

(1) Confidentiality of privacy data: Due to the distributed characteristics of SWIM, the data are distributed in each service node, users cannot guarantee whether the privacy data will be leaked, and this hinders the further development of SWIM.

(2) User access transgression: SWIM data comes from all kinds of ATM systems worldwide and has strong liquidity. After multiple transmission of data, information copy may be formed in unauthorized user nodes, so there is a risk of user access penetration.

(3) Illegal user access: SWIM uses the name as the identification of the service information. If the external users master the naming rules, they can initiate access to the corresponding information, which may cause the leakage of private data.

ATM data of all kinds need complete information interaction between systems through SWIM at each stage of aircraft operation. According to the above security risk analysis, many security threats against SWIM data will affect the normal operation of the aircraft, and even threaten the safe operation of civil aviation system in serious cases.

4. SWIM Access Control Scheme

XACML is the most effective and appropriate access control policy language because it is compatible with different platforms and provides a distributed and flexible method for various access control scenarios of different systems [25]. XACML provides a combination algorithm, which combines a decision or the influence of multiple policies or rule elements into a decision through the policy combination algorithm of the policy element and the rule combination algorithm of the rule element and finally grants the applicant

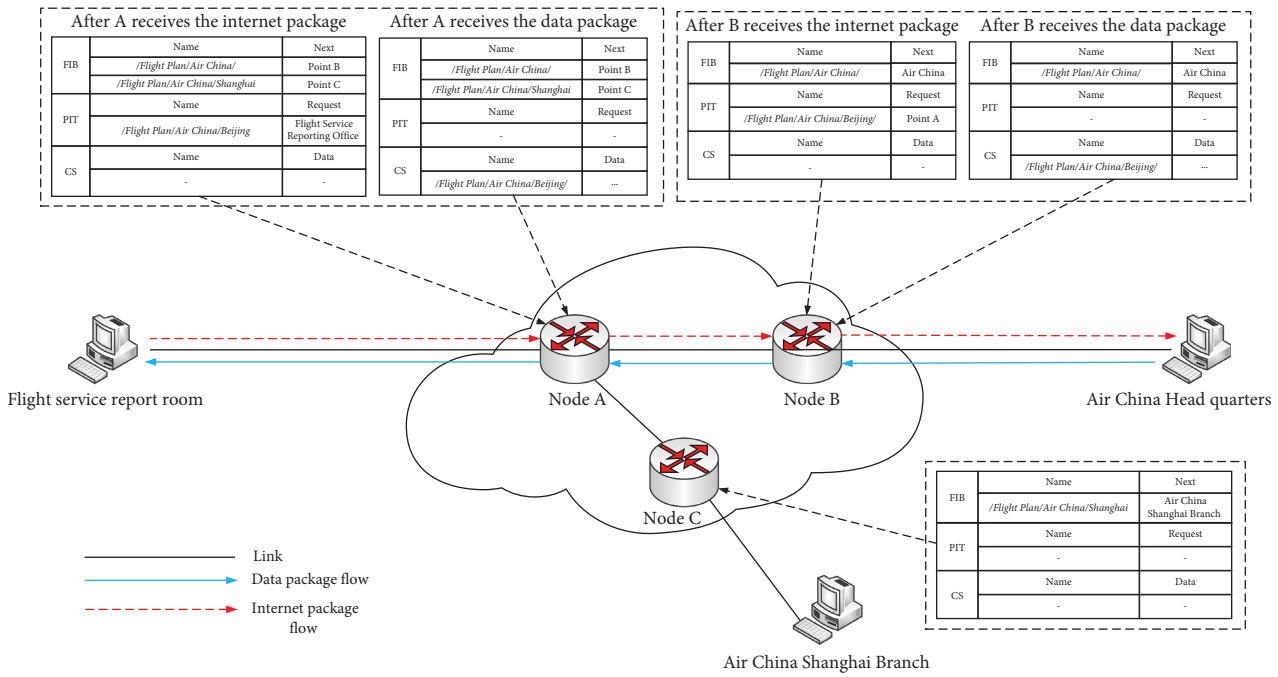


Figure 2: SWIM information routing mechanism.

authority [26]. On the basis of the characteristics of SWIM and the actual business process of civil aviation in China, an ABAC scheme is proposed. In accordance with the improved XACML authorization model, an attribute association method is designed in the scheme to support more capable expression authorization policies and to enable flexible, fine-grained access control.

ABAC is a milestone in the development of access control technology, and XACML is the most influential access control policy expression language [27]. ABAC has gained increasing popularity in the last years due to its flexibility and expressiveness [28]. On the basis of the characteristics of SWIM, a flexible and fine-grained access control scheme based on ABAC and XACML is proposed to ensure the security of private data in SWIM.

4.1. SWIM Access Control Framework. SWIM has the characteristics of distribution and open network. It contains massive aviation data. In SWIM, user nodes are constantly moving, and accessible resources are updated in real time [15]. ABAC supports dynamic access control management by changing attribute values without changing the subject-object relationship that defines the underlying access control policy [29]. An ABAC rule is a tuple that specifies a collection of users, objects, operations, and constraints involving user and object properties [30]. In ABAC, a principal's request to perform an action on an object is granted or denied based on the principal's assigned attributes, the object's assigned attributes, environmental conditions, and a set of policies specified based on those attributes and conditions [31]. In this chapter, we propose the SWIM access control framework.

The overall framework of the access control scheme is shown in Figure 3, which mainly includes policy execution point (PEP), policy decision point (PDP), policy administration point (PAP), and attribute authority (AA). The PEP, PDP, PAP, and AA are responsible for the context conversion of the access request; determining whether the request conforms to the corresponding authorization policy and returns the determination result; managing the authorization policy of various service information; and managing various attributes of SWIM, respectively.

The service requester finds the SWIM service node containing the request information or the corresponding copy through the FIB of each node in the transmission path and transmits the original access request (NAR) to the PEP of the node. On the basis of the access request information in the NAR, the PEP requests various types of attributes from the AA, uses these attributes to construct an attribute-based access request (AAR), and transmits it to the PDP. The PDP requests the matching authorization policy from the PAP in accordance with the naming of the service information requested in the AAR, determines the access authority by using the attribute information in the AAR and the association relationship of the access structure in the policy, and finally returns the determination result to the PEP. If the result of the determination is that access is permitted, the PEP returns the corresponding service information from the local CS to the requester; if the authority determines that the access is denied, the PEP returns the corresponding failure information.

4.2. Improved XACML Authorization Model. Formulation of an authorization policy is a prerequisite for implementing access control. XACML is a typical descriptive language in the ABAC environment [32]. We use the XACML language and the ABAC model to create access control policies [33]. XACML is based on XML and has good versatility,

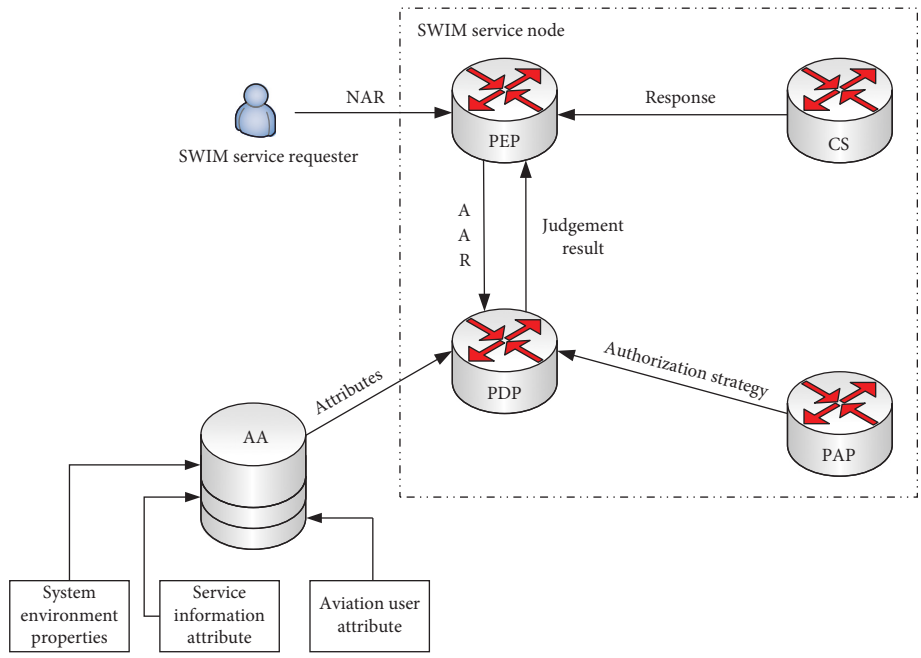


Figure 3: Access control framework.

scalability, and platform independence. It describes access control requests and policies through attributes in subjects, resources, behaviors, and environmental information. Rules in XACML evaluate incoming access requests based on conditions passed using attribute expressions. However, XACML has a complex structural hierarchy where security policies are very flexible, and the range of attribute values defined by the policies is likely to overlap [34].

As a complete access control mechanism, XACML provides a flexible set of policy representation methods. With <Policy> as the root element of the policy, two child elements <Target> and <Rule> are defined below. <Target> indicates the target, which can be divided into the target targeted by the overall strategy and the target targeted by the location rule. <Rule> represents the rule and is an important part of the strategy. Each strategy can consist of one or more rules, and each rule needs to build an application environment together by using <Target> and <Condition>. The various attribute information of the two can be represented by four elements: <Subject>, <Resource>, <Action>, and <Condition> [35].

To enhance the ability of the strategy to express, the original XACML sets the child element <Subject> in the subject element <Subjects> for implementing the OR operation between the subject users. Hence, one XACML access policy can control multiple different subjects. While this approach enhances flexibility, the same attributes that exist between different users generate redundant information, resulting in increased storage overhead.

This study tests the storage space occupied by multiple groups of access policies with different attribute numbers and takes the average value. The experimental results are shown in Figure 4. As the number of attributes in the policy increases, the proportion of the space occupied by the attribute set in the policy increases. When the policy contains 10 attributes, the attribute set accounts for 80.3% of the total storage space. When the attribute exceeds 25, the proportion reaches 90.8%. Therefore, the attribute set is the main component of the strategy. Decreasing the redundant attribute information can effectively reduce the storage space of the policy.

Given that the cache-and-forward mechanism is the basis for SWIM to implement efficient information interaction, the storage capacity of CS in the service node directly affects the information transmission efficiency in the system. To allocate as much space as possible to the CS in exchange for enhanced transmission efficiency, the original XACML authorization model is improved. The improved model is shown in Figure 5.

In this scheme, the element in the section describing "users" in XACML policy description language is improved, and four attributes such as Group, Threshold, Extern Group, and Extern Threshold are added to the element. The current packet, the threshold value of the current packet, the external packet, and the threshold value of the external packet are, respectively, represented in the improved XACML authorization model. During XACML access control, different subpolicies are grouped to establish connections between these subpolicies, and external department limits are set according to access policies. Finally, Group, Threshold, Extern Group, and Extern threshold of each attribute in the policy are assigned by Group. By introducing these attributes and using the concept of threshold, the logical operation mode in the policy is expanded and the expression ability of the policy is enhanced.

In addition, XACML specifies four combination algorithms [36] to resolve the policy problem of the same resource managed by different PAPs. The rejection priority

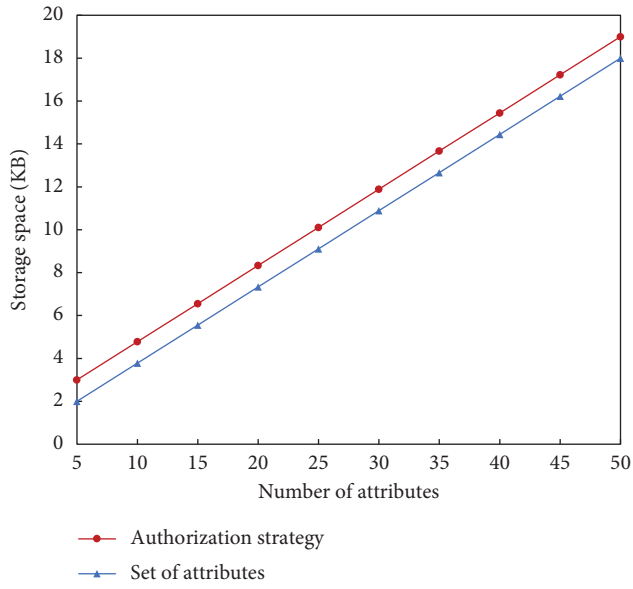


Figure 4: Strategy storage space comparison chart.

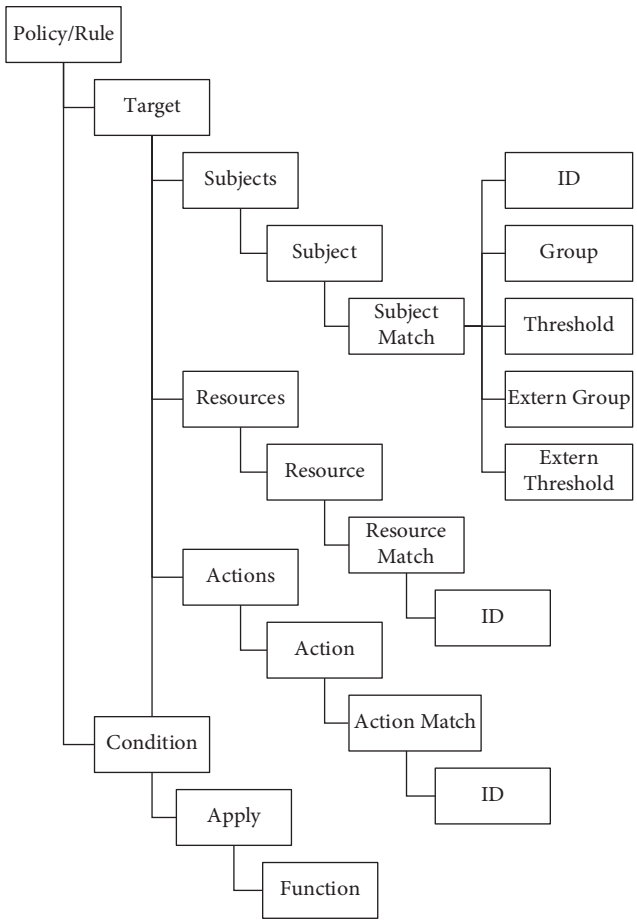


Figure 5: Improved XACML authorization model.

algorithm is adopted in the scheme. If a rule or policy application obtains a rejection result, then the user does not have the permission. The scope of the authority is further narrowed to ensure the security of service information.

The improved policy description method realizes flexible transformation between XACML and Boolean expression and can combine multiple access policies of the same service information into one description, which has good expression ability. The above improvements led to the optimization of ABAC model, which simplified the decision point for strategy traversal process, improved the efficiency of the matching attributes, enriched the access strategy expression type, effectively saved the storage space, and improved the whole performance of this scheme. This method is also the basis of attribute matching algorithm in subsequent chapters.

4.3. Attribute Association Algorithm. Attribute association is a process of associating and matching the attribute set possessed by the requester with the access structure in the corresponding resource strategy and obtaining the determination result. It is an important step for authority determination. In this scheme, a linear secret sharing scheme (LSSS) matrix is difficult to design for its abstract mathematical structure [37], but it has good application in attribute association. To express the LSSS matrix structure, LSSS and the monotone span program (MSP) are introduced and combined with the improved XACML to design a flexible attribute association algorithm.

4.3.1. Basic Theory. If a secret sharing scheme $\prod$ for a set of participants $P = \{P_1, P_2, \ldots P_n\}$ is linear over a finite field Z_P, then the following properties should be met [38]:

(1) Everyone is distributed a share Z_P [39].

(2) There is a shared generation matrix A for the sharing scheme $\prod$. Suppose l rows and n columns exist in A, for all $i = 1, 2, \ldots, l$, the first row i of the matrix A is marked as a participant by $\rho(i)$. $\rho(i)$ is a mapping from $\{1, 2, \ldots, l\}$ of participant set P. At the same time, there is an n-dimensional column vector $v = (s, r_2, \ldots, r_n)^T$. $s \in Z_P$ is the secret value to be shared, and $r_2, r_3, \ldots, r_n \in Z_P$ is a random number. Then, Av is the secret sharing value in the secret sharing scheme $\prod$, and the secret value $(Av)_i$ belongs to the participant $\rho(i)$.

All the LSSS follow the above definition and have corresponding secret recovery algorithms. If the scheme $\prod$ is for the LSSS of the access structure Λ, for the authorization set $S \in \Lambda$, let $I = \{i: \rho(i) \in S\}$; a set of recovery coefficients $\{\mu_i\}_{i \in I}$ can be calculated to make $\sum_{i \in I} \mu_i M_i = (1, 0, \ldots, 0)$, such that $\sum_{i \in I} \lambda_i v = \sum_{i \in I} \mu_i M_i v = s$ can be calculated and the original secret value can be restored [40].

Calculating the recovery coefficient μ_i is essentially a linear equation solving problem. μ_i can be found in polynomial time associated with the size of the shared secret matrix A. For unlicensed collections, there is no corresponding recovery coefficient, which guarantees the security of LSSS.

4.3.2. Algorithm Design. In this scheme, the SWIM attribute corresponds to the participant in the LSSS concept, and the access right corresponds to the secret value to be shared. That is, each row vector M_i in the secret sharing matrix M is

marked as an attribute by $\rho(i)$, and the authority information is scattered into attributes in the form of secret shared values. If the user's attribute set satisfies the access structure in the corresponding policy, then the user can recover the authority value through the secret recovery algorithm and obtain the corresponding operation authority.

The implementation flow of the attribute association algorithm is shown in Figure 6, which mainly includes three steps: structural transformation, secret sharing matrix generation, and authority determination. This section elaborates the algorithm steps.

(1) Structural Transformation. The transformation structure is mainly for two aspects. They are the access structures of the attribute set in the access request and of the access control policy corresponding to the subscribed service. The original tree structure should be converted into a standard Boolean type. A Boolean expression is a way to describe a logical relationship by using attribute and threshold values. It is concise and flexible and is the basis for subsequent steps.

For example, the flight plan service strategy issued by Air China stipulates that the intelligence service of the flight service report room of the North China Air Traffic Control Administration and the executives within Air China can subscribe. The corresponding strategy is converted into a standard Boolean strategy:

$$P_{R_{FP}} = \begin{pmatrix} (\text{CAAC North China Regional Administration}, \\ \text{Flight Service Reporting Office}, 2), \\ (\text{Air China}, \text{Senior Manager}, 2), 1. \end{pmatrix}. \tag{1}$$

(2) Generation of Secret Sharing Matrix. After the previous step of structural transformation, the Boolean expression of the access strategy is obtained. In this step, the MSP is used to process the strategy further, generate a secret sharing matrix, map each row vector in the matrix to the attribute, and form an access structure in the form of a matrix and a mapping finally.

Through the logical relationship of Boolean expressions, the strategy can be decomposed into several subelements. This method will transform the strategy layer by layer in units of subelements. The child element can either be a decomposed subpolicy or an attribute value. The MSP uses the threshold $(t, n), t \le n$ to convert the child elements into a matrix form [41], where t represents the threshold value and n represents the number of attributes in the child element. A generation matrix for a threshold (t, n) can be expressed as

$$M_{(t,n)} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 2^2 & \dots & 2^{t-1} \\ 1 & 3 & 3^2 & \dots & 3^{t-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & n & n^2 & \dots & n^{t-1} \end{pmatrix}. \tag{2}$$

Suppose the original matrix is $M_1 = (M_{1P}/M_1)$, where M_{1P} expresses the subelements decomposed in this operation mapped to the corresponding row vectors in the matrix. Let $M_{1P} = \begin{pmatrix} v_1 & v_2 & \dots & v_q \end{pmatrix}^T$, $v_1, v_2, \dots, v_q$, represent the row vector. Owing to the layer-by-layer decomposition, there is only one row vector $M_{1P} = (v_1)$ in this method.

Assuming that the matrix M_2 is generated by the MSP on the basis of the threshold (t, n), let $M_2 = \begin{pmatrix} U & \overline{M_2} \end{pmatrix}$. U is the first column element of M_2, which can be written as $U = \begin{pmatrix} u_1 & u_2 & \dots & u_n \end{pmatrix}^T$. Let $M' = \begin{pmatrix} u_1 \cdot v_1 & u_2 \cdot v_1 & \dots & u_n \cdot v_1 \end{pmatrix}^T$, and the result of each recursion is

$$M = \begin{pmatrix} M' & \overline{M_2} \\ \overline{M_1} & 0 \end{pmatrix}. \tag{3}$$

The above operation is looped until all the child elements in the expression are decomposed into a single attribute value, and the corresponding secret sharing matrix M and its mapping ρ can be obtained. After $P_{R_{FP}}$ conversion in the previous step, the corresponding results are as follows:

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \end{pmatrix},$$

$$\rho = \begin{pmatrix} \text{CAAC North China Regional Administration} \\ \text{Flight Service Reporting Office} \\ \text{Air China} \\ \text{Senior Manager} \end{pmatrix}. \tag{4}$$

4.4. Performance Analysis. This section mainly analyzes the attribute association algorithm and overall performance in the scheme.

As the number of attributes increases, the structure of the access policy will be more complicated, and the attribute association time will also increase. The relationship between the permission judgment time and the number of attributes of the attribute association algorithm is verified. The experiment uses a SWIM user with 5 attributes to initiate 100 subscription requests for the service corresponding to the access policy with 5 to 50 attributes, records the permission determination time, and calculates the average time. Table 3 shows the test results. The permission judgment time increases with the increase in the attribute value. When the quantity of attributes increases from 5 to 50, the time only increases by nearly 9 ms, and the growth trend is inevident.

Given that SWIM is still in the construction stage, no specific technical indicators exist for access control functions. To verify the performance of the solution, this section compares the open-source XACML solutions provided by Sun and tests the time of the attribute association algorithm

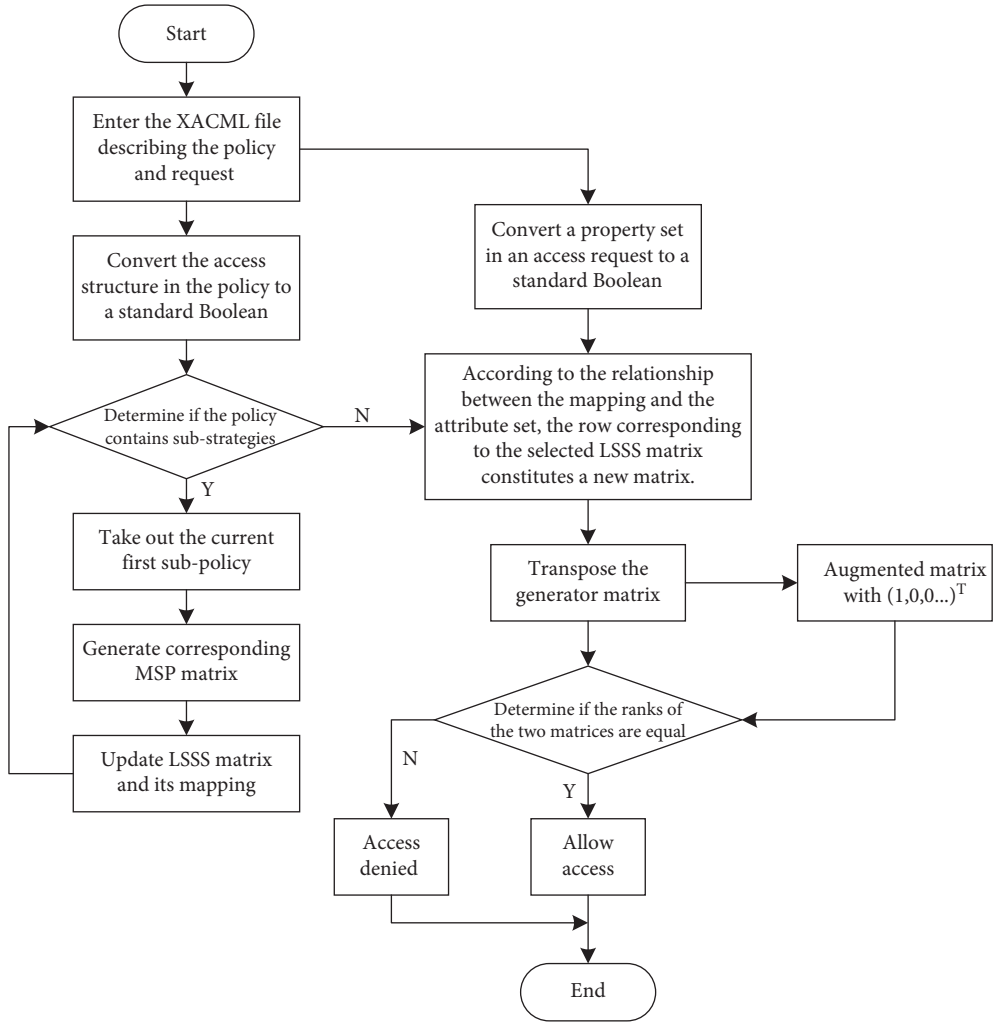


Figure 6: Flowchart of the attribute association algorithm.

and the overall solution. The experimental method is the same as the above experiment, which uses the same user to subscribe to the service corresponding to the access policy containing different attribute numbers multiple times and records the corresponding time. The experimental results are shown in Figure 7.

Figure 7(a) shows the performance comparison of the attribute association methods in the two schemes. As the LSSS and the MSP are introduced into the scheme, the matching speed is slower, and it is more time-consuming than the original scheme.

However, the slopes of the two curves in the figure demonstrate that the two schemes have similar growth rates with the increase in the quantity of attributes. The strategies in each of the two schemes add 45 attributes, and the time increases by 8.876 and 8.816 ms. Therefore, the scheme does not cause the system performance to deteriorate due to the increase in the quantity of attributes.

Figure 7(b) shows a comparison of the overall performance of the two schemes. The time overhead of this solution is significantly smaller than the original scheme, and this solution has better performance. Although the scheme consumes more time in the attribute association method, the time cost of the overall scheme is similar to the method itself. The original scheme also has preorder work, such as policy traversal, which greatly reduces the overall performance of the scheme.

Compared with the original XACML access control scheme, this scheme designs an attribute association method based on LSSS in the permission determination process, which increases the attribute association time compared with the total access control time of the original scheme, so the time complexity increases. However, the attribute association algorithm designed in this scheme, combined with the improved XACML policy description language, can support access policies with richer expressive capabilities, optimize the performance of policy expression, and make access policies correspond to SWIM service information. The SWIM system involves a large number of civil aviation business applications. The civil aviation business data are shared through the subscription and publishing function, which consumes a lot of memory resources and increases the space complexity. However, the SWIM subscription and publishing function is not sensitive to time complexity.

Table 3: Permission judgment time (ms).

Attribute number	Decision number							
	1	2	3	...	98	99	100	Average value
5	35.446	35.290	35.411	...	35.637	35.157	35.487	35.365
10	36.715	35.869	36.125	...	35.955	35.830	36.514	36.154
15	38.030	37.467	37.820	...	37.401	37.212	37.401	37.718
20	38.949	38.220	38.688	...	38.029	38.318	38.923	38.560
25	38.768	39.414	39.830	...	39.771	40.070	40.009	39.622
30	40.758	40.317	41.300	...	40.291	41.247	40.729	40.661
35	41.747	42.520	42.302	...	42.011	41.747	41.735	41.885
40	42.607	42.934	43.105	...	43.062	42.829	42.486	42.860
45	43.897	43.206	43.289	...	43.219	42.886	44.100	43.386
50	43.861	44.957	44.274	...	44.025	44.100	44.618	44.241

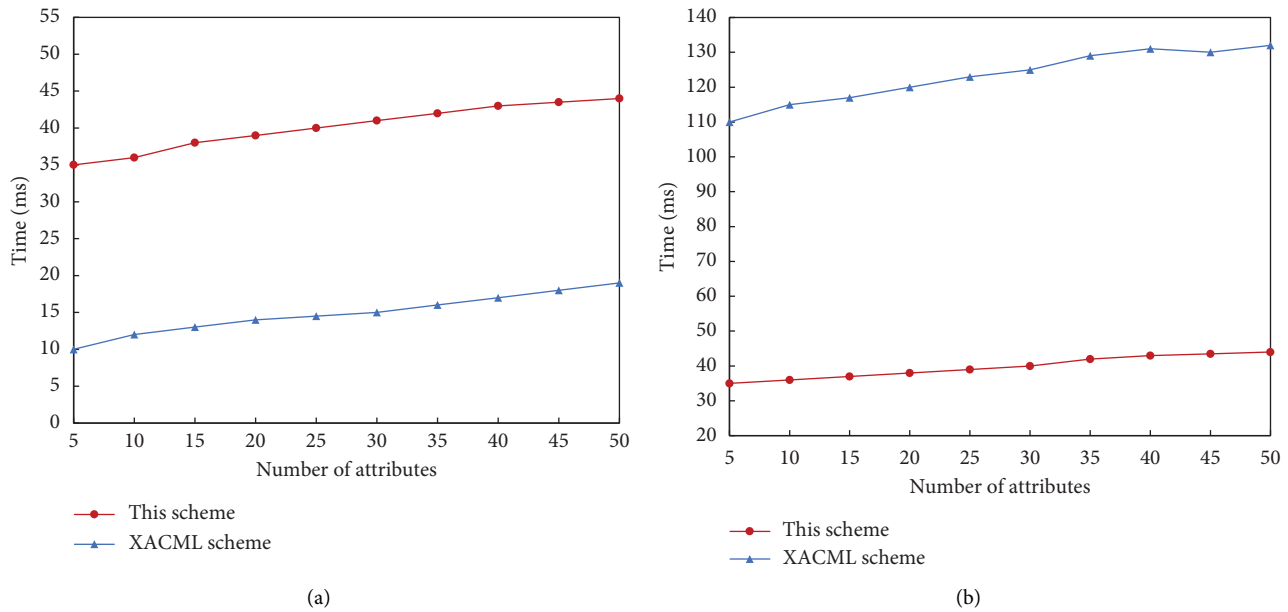


Figure 7: Performance comparison. (a) Performance comparison of attribute association algorithms. (b) Overall program performance comparison.

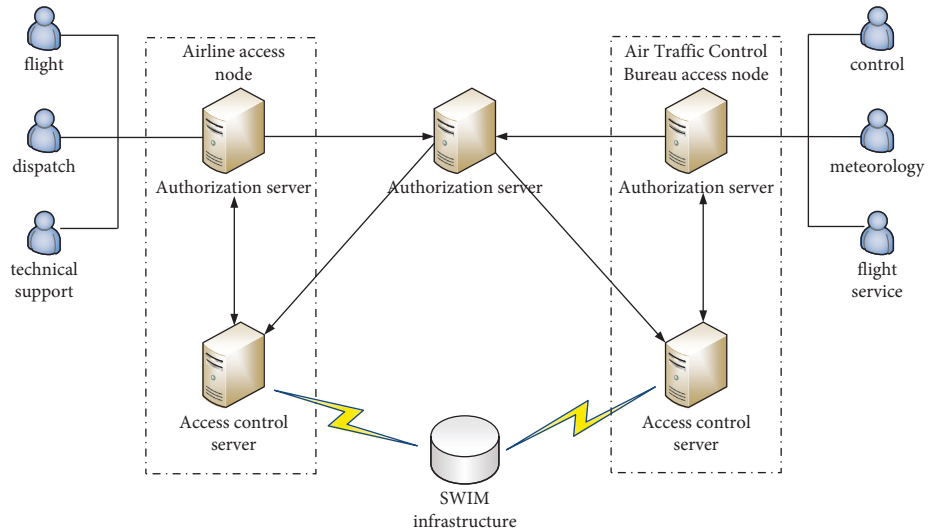


Figure 8: Experimental environment diagram.

There may be a certain time delay in shared data, and the priority of space complexity is higher than that of time.

5. Simulation Experiment and Result Analysis

5.1. Experimental Environment. On the basis of SWIM's architectural features and business processes, an experimental environment is built, which deploys access control solutions to the system and tests them. Figure 8 shows the network topology of the experimental environment.

The experiment simulates the information interaction between multiple departments of the two SWIM access nodes of the airline and the Air Traffic Control Bureau. The service information involved is derived from the real civil aviation network.

5.2. Stress Testing. With the stress test tool LoadRunner 11.0, the overall program performance is evaluated. It compares the test results with the original XACML solution to evaluate whether the solution is suitable for SWIM simulation environment. The test cases are shown in Table 4.

In accordance with the SWIM system architecture, the access control function is deployed on each access node. Therefore, the test environment of the performance indicator should be based on the actual operating environment of the node. Given that some copies of service information are stored on each node, the number of concurrent accesses of resources by users to the same node is effectively alleviated. Combined with the basic configuration of the experimental platform, the performance of the solution is tested in an environment where 100 users access the same type of service information concurrently.

The initial number of users in the test script is 20, and 20 users are added every 10 s until 3 min after the increase to 100 users. Then, 20 users are reduced every 10 s until all the users in the system are logged off. The average transaction response time and CPU usage are the main indicators in the test.

Figure 9 shows the average transaction response time of the two schemes during the test. The average transaction response time increases with the quantity of concurrent users. When the user reaches the peak, the response time of the two schemes is stable at 0.163 and 0.121 s. Compared with the original scheme, the average response time of the proposed scheme is reduced by 0.042 s, and the maximum response time is reduced by 0.064 s. Therefore, the scheme is slightly faster than the original scheme in execution speed.

Figure 10 shows the CPU usage of the two schemes during the test. As the quantity of concurrent users increases, the CPU usage also increases. When the concurrent amount reaches the peak, the CPU usage of the two schemes is stable at 32.046% and 35.847%. The proposed scheme has better performance than the original XACML scheme.

In accordance with the number of transaction passes and the number of transaction failures in the system's stress test statistics, the success rate of this solution can reach 99.75%. This finding indicates that the scheme can be used normally in a relatively conserved SWIM environment.

Lastly, the performance of the two schemes in different concurrent environments is compared by comparing the number of users in the above test cases. The comparison results are shown in Figure 11.

Figure 11 depicts that when the quantity of users increases from 50 to 200, the average response time of the two schemes increases with the number of users; when the number of users increases from 200 to 500, the average response time of the two schemes stabilizes at 0.160 and 0.198 s. The reason is that the bandwidth of the experimental platform is limited, and the total amount of transactions tested does not increase indefinitely. When the user reaches a certain amount, the throughput, the number of clicks, and the total amount of transactions are at fixed values, resulting in stable response time. In accordance with the curve change in the figure, the performance of the proposed scheme is better than that of the original XACML scheme in the rising phase with lower concurrency and in the stable phase with larger concurrency.

6. Security Analysis

The security of this program is mainly analyzed from the following aspects:

Availability: In accordance with the routing characteristics in SWIM, users only need to clarify the service information naming rules in the system without having to know the specific location of the requestor or the publisher in the network. This scheme uses XACML to describe the AAR and access policy and utilizes the service information naming as the identifier of the object resource. Therefore, it only needs to obtain the requester attribute set and the requested information naming to complete the determination of the authority, satisfy the SWIM routing features, and ensure the availability of service information.

Confidentiality: This scheme is based on the fact that the user has passed the security certification. The user's attribute set has been verified by a trusted third party and managed by the AA. In this case, the user cannot forge the attribute information, and the attributes carried in the access request are all true.

The threshold (t, n) combines the secret value s and $n - 1$ random number into n rank vector and multiplies it with the multistep n coefficient row vector to obtain a secret share value, which is distributed to multiple users. The secret value can be reconstructed only if t or more than t users jointly decrypt by using part of the secret information held.

The attribute association algorithm of the scheme is based on the threshold secret sharing algorithm, and the permission value is dispersed into each attribute. Only when the number of attributes of the user reaches the threshold value and the user is determined as an authorized user, the confidentiality of

Table 4: System test cases.

Test case number	SWIMclient_test_01
Testing purposes	Overall performance of two access control schemes when validating large numbers of users concurrently accessing specific resources
Test conditions	The user's attribute set, and resource which can satisfy the user's permission to access specific resources
Script description	Enter the service information type and perform the access operation
Testing scenarios	Concurrent access to meteorological information in Beijing using 100 virtual users for two access control schemes
Test result description	In the test scenario, the user can access the specific service information normally, and the performance of the solution meets the expected design indicators.

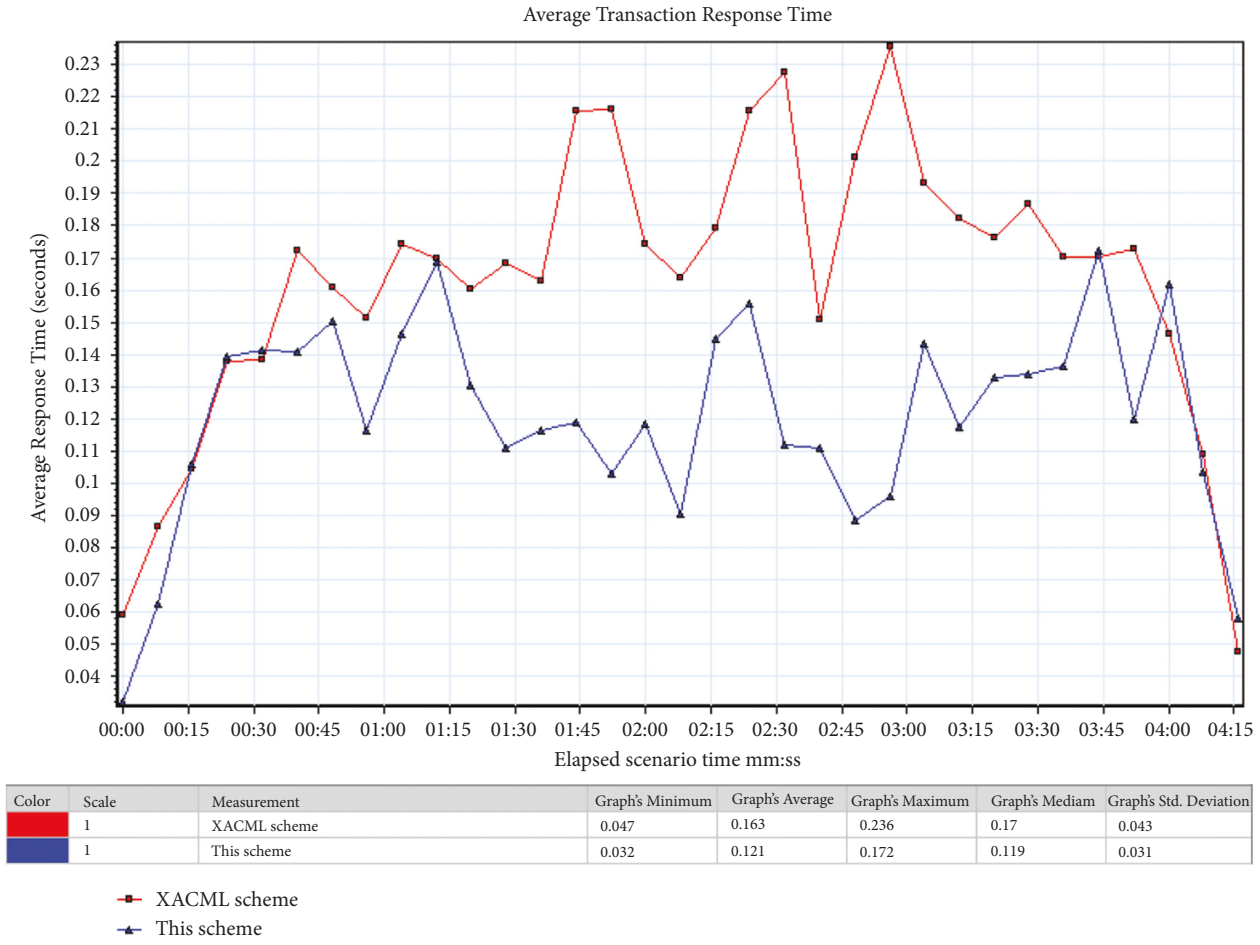
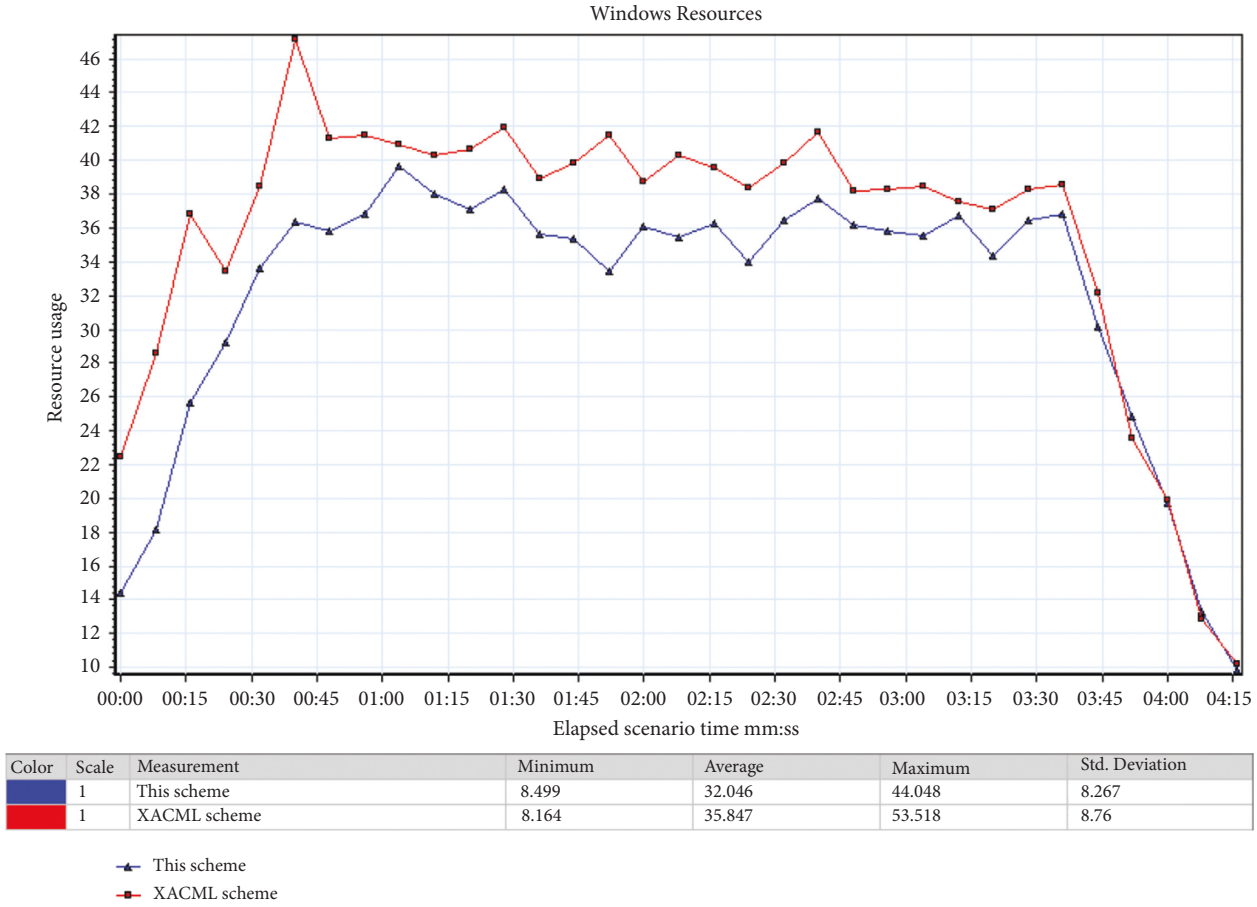


Figure 9: Comparison of average transaction response time.

the privacy information in the system can be guaranteed.

In addition, as multiple users cannot aggregate attributes, unauthorized users cannot collude to obtain access rights. Given that the permission value uses the system time as the seed value for each judgment, the permission value is randomly generated, and the attacker cannot obtain SWIM data from the previously intercepted permission value.

Integrity: Owing to the flexible naming scheme in SWIM, the elements in the name are not required to be human readable, such that message digest can be added to the URL naming rules. Each user generates local digest only following the Hash algorithm specified in SWIM. Comparing the local digest and information naming can ensure that the information has not been tampered during the transmission process. Information integrity protection in SWIM is then realized.



Color	Scale	Measurement	Minimum	Average	Maximum	Std. Deviation
	1	This scheme	8.499	32.046	44.048	8.267
	1	XACML scheme	8.164	35.847	53.518	8.76

Figure 10: CPU usage comparison chart.

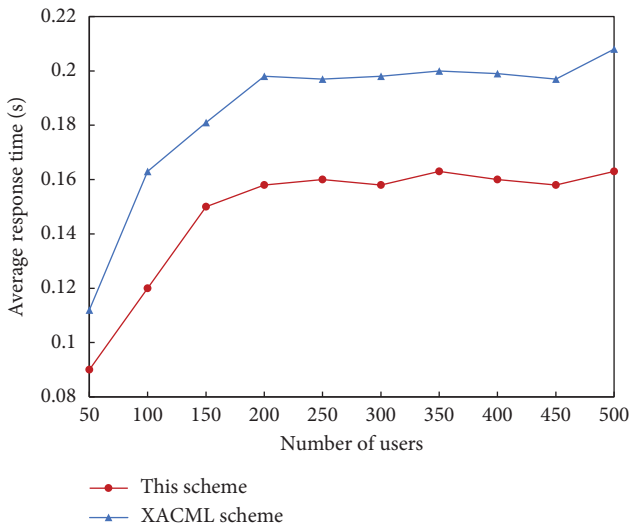


Figure 11: Comparison of average response time in different user scenarios.

7. Conclusion

In consideration of the business process in China's civil aviation, the characteristics of SWIM, and its security requirements, an ABAC scheme for SWIM environment is proposed. This scheme combines the improved XACML to design an access control framework. On this basis, an attribute association method is designed to support a strategy with enhanced representation capability. The performance test results show that compared with the original XACML, although the attribute association algorithm in the proposed scheme takes a long time, the overall performance is improved and some of the policy storage space is saved. The stress test results show that the technical indicators in the proposed scheme are superior to those in the original XACML. However, the attribute association algorithm in this article essentially uses the time complexity to exchange the space complexity. In terms of the algorithm alone, the proposed scheme takes more time than the original scheme. In the future, the efficiency of the algorithm should be improved by simplifying the algorithm flow and optimizing the algorithm structure.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant nos. U1933108, 62172418, and U2133203), the Scientific Research Project of Tianjin Municipal Education Commission (Grant no. 2019KJ117), and the Open Fund of Key Laboratory of Airworthiness Certification Technology of Civil Aviation Aircraft (SH2021111907).

References

[1] Z. Liu, "In domain user identity authentication method for system wide information management," in *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, pp. 479–482, IEEE, Guangzhou, China, May 2020.

[2] X. Lu, K. Morioka, T. Koga, and Y. Sumiya, "Air-ground system wide information management to achieve," in *Proceedings of the Safe Flight Operation 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 42–49, IEEE, Hangzhou, China, January 2019.

[3] Faa, "NextGen implementation plan 2018-19," https://www.faa.gov/nextgen/media/media/NextGen_Implementation_Plan-2018-19.pdf.

[4] Z. Wu, S. Zhou, and Y. Meng, "Research on SWIM services dynamic migration mechanism," in *Proceedings of the IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, pp. 1035–1039, IEEE, Athens, Greece, August 2018.

[5] Z. Wu, T. Zhao, and L. Jin, "Authentication method in SWIM based on improved Diameter/EAP-MD5," *Journal on Communications*, vol. 35, no. 8, pp. 1–7, 2014.

[6] W. Zhijun, C. Zihan, W. Caiyun, and L. jin, "Access control scheme with attribute revocation for SWIM," *The Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 6, pp. 49–54, 2017.

[7] A. Liu, X. Du, and Na Wang, "Big data Access control based on block chain," *Journal of Software*, vol. 30, no. 9, pp. 2636–2654, 2019.

[8] R. Du, Y. Liu, and J. Tian, "Access control method based on smart contract in Internet of things," *Journal of Computer Research and Development*, vol. 56, no. 10, pp. 2287–2298, 2019.

[9] Z. Zhang, *"Based on Block Chain between the Domain of Property Research and Implementation of Access Control"*, Beijing university of technology, 2020.

[10] J. Y. Wu, *"Research and Implementation of Risk-Based XACML Access Control Model under Fog Computing"*, Nanjing university of posts and telecommunications, 2021.

[11] D. Han, W. Yuan, X. Duan, and L. Zhang, "An Approach to XACML policy query based on Attribute and or Matrix and type Analysis," *Computer science*, vol. 45, no. 09, pp. 224–229, 2018.

[12] C. Wang, R. Li, X. Gu, and J. Tang, "Policy conflict detection and resolution based on XACML," *Computer science and exploration*, vol. 12, no. 01, pp. 1–16, 2018.

[13] Y. Luo, Q. Shen, and Z. Wu, "A metamodel based access control policy description language," *Journal of Software*, vol. 31, no. 02, pp. 439–454, 2020.

[14] R. Khaled and J. Cheng, "Adaptive XACML access policies for heterogeneous distributed IoT environments," *Information Sciences*, vol. 548, pp. 135–152, 2021.

[15] I. Wilson and S. Yang, "Security for system wide information management," in *Proceedings of the 2017 Integrated Communications, Navigation and Surveillance Conference*, pp. 1–13, IEEE, Herndon, USA, 2017.

[16] M. Moallemi, C. A. Castro-Pena, M. Towhidnejad, and B. Abraham, "Information security in the aircraft access to system wide information management infrastructure," *2016 Integrated Communications Navigation and Surveillance (ICNS)*, vol. 1A3-1-1A3-7, 2016.

[17] C. W. Johnson, *"Cyber Security and the Future of Safety-Critical Air Traffic Management: Identifying the Challenges under NextGen and SESAR", 10th IET System Safety and Cyber-Security Conference*, pp. 1–6, IET, Bristol, 2015.

[18] J. Hird, M. Hawley, and C. Machin, *"Air Traffic Management Security Research in SESAR", 11th International Conference on Availability, Reliability and Security*, pp. 486–492, IEEE, Salzburg, 2016.

[19] *SWIM Concept – Draft Version 0.9, ICAO Air Traffic Management Requirements and Performance Panel*, (ATMRPP), 2013.

[20] De Haan and Johannes, "Specific air traffic management cybersecurity challenges: architecture and supply chain," in *Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pp. 245–249, ACM, Seoul, Korea, 2020.

[21] J. Robb, "System wide information management (SWIM): program overview and status update," in *Proceedings of the 2014 Integrated Communications Navigation and Surveillance Conference*, pp. 1–15, IEEE, Herndon, USA, October 2014.

[22] S. Egami, X. Lu, T. Koga, and Y. Sumiya, "Enriching geospatial representation for ontology-based aviation information exchange," in *Proceedings of the 2019 IEEE 8th Global Conference on Consumer Electronics*, pp. 238-239, IEEE, Osaka, Japan, 2019.

[23] Y. Sun, Yu Zhang, and H. Zhang, "Review of research on information center network architecture," *Chinese Journal of Electronics*, vol. 44, no. 8, pp. 2009–2017, 2016.

[24] G. Xylomenos, C. N. Ververidis, V. A. Siris, and N. C. X. K. V. G. C. Fotiou, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[25] L. A. Charaf, I. Alihamidi, A. Addaim, and A. A. Madi, "A distributed XACML based access control architecture for IoT systems," in *Proceedings of the 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology*, pp. 1–5, IEEE, Meknes, Morocco, April 2020.

[26] S. Haguouche, Z. Jarir, and K. Yeh, "Towards a secure and borderless collaboration between organizations: an automated enforcement mechanism," *Security and Communication Networks*, vol. 2018, no. 4, pp. 1–13, 2018.

[27] X. Luo and S. Wang, "Improved access control decision diagrams for ABAC policy evaluation and management," in *Proceedings of the2019 6th International Conference on Systems and Informatics*, pp. 932–937, IEEE, Shanghai, China, November 2019.

[28] C. Morisset, T. A. C. Z. Willemse, and Absac, "Attribute-based access control model supporting anonymous access for smart cities," *Cybersecurity*, vol. 2, no. 1, pp. 25–31, 2019.

[29] H. Kim, D.-K. Kim, and A. Alaerjan, "ABAC-based security model for DDS," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, 2021.

[30] S. Chakraborty, R. Sandhu, and R. Krishnan, "On the feasibility of attribute-based access control policy mining," in *Proceedings of the 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 245–252, IEEE, LosAngeles, USA, September 2019.

[31] B. Gunjan, A. Vijayalakshmi, V. Jaideep, and S. Shamik, "Securing the cloud resources using attribute based access control," *Journal of Computer Security*, vol. 27, no. 4, pp. 483–506, 2019.

[32] N. I. U. De hua, M. A. J. feng, and M. A. Zhuo, "Attribute-based security enhanced cloud storage access control scheme," *Transactions of Communications*, vol. 23, no. 45, pp. 276–284, 2013.

[33] H. Razouki, "Dynamic data protection for mobile agents: XACML/ABAC policies," in *Proceedings of the Digital Technologies and Applications. Proceedings of ICDTA 21. Lecture Notes in Networks and Systems*, pp. 483–493, IEEE, London,UK, June 2021.

[34] G. Liu, W. Pei, Y. Tian, C. Liu, and S. Li, "A novel conflict detection method for ABAC security policies," *Journal of Industrial Information Integration*, vol. 22, no. 8, pp. 100200–100245, 2021.

[35] K. Seol, Y.-G. Kim, E. Lee, Y.-D. Seo, and D.-K. Baik, "Privacy-preserving attribute-based access control model for XML-based electronic health record system," *IEEE Access*, vol. 6, pp. 9114–9128, 2018.

[36] R. Pan and G. M. Wang, "An attribute-based access control policy retrieval method based on binary sequence," *Security and Communication Networks*, vol. 2021, pp. 1–12, 2021.

[37] J. Zhao and H. Gao, "LSSS matrix-based attribute-based encryption on lattices," in *Proceedings of the 2017 13th International Conference on Computational Intelligence and Security*, pp. 253–257, IEEE, Hong Kong, China, February 2017.

[38] H. Wang, N. Shu, C. Wang, and J. Chen, "A novel linear secret sharing scheme with multiple prime moduli," in *Proceedings of the 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 1689–1693, IEEE, Guilin, China, 2017.

[39] Y. Li, H. Wang, S. Wang, and Y. Ding, "Attribute-based searchable encryption scheme supporting efficient range search in cloud computing," in *Proceedings of the 2021 IEEE Conference on Dependable and Secure Computing*, pp. 1–8, IEEE, Fukushima, Japan, February 2021.

[40] K. K. Phiri, "An efficient compartmented secret sharing scheme based on linear homogeneous recurrence relations," *Security and Communication Networks*, vol. 63, pp. 32–44, 2019.

[41] S. Wang, J. Zhou, J. K. Liu, J. Yu, and J. W. Chen, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265–1277, 2016.