Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042305

ARTICLE

Cross-Domain Authentication Scheme Based on Blockchain and Consistent Hash Algorithm for System-Wide Information Management

Lizhe Zhang[1,2,*], Yongqiang Huang[2], Jia Nie[2] and Kenian Wang[1,2]

[1]Key Laboratory of Civil Aircraft Airworthiness Technology, Civil Aviation University of China, Tianjin, 300000, China

[2]School of Safety Science and Engineering, Civil Aviation University of China, Tianjin, 300000, China

*Corresponding Author: Lizhe Zhang. Email: lzzhang@cauc.edu.cn

Received: 25 May 2023 Accepted: 12 August 2023 Published: 29 November 2023

ABSTRACT

System-wide information management (SWIM) is a complex distributed information transfer and sharing system for the next generation of Air Transportation System (ATS). In response to the growing volume of civil aviation air operations, users accessing different authentication domains in the SWIM system have problems with the validity, security, and privacy of SWIM-shared data. In order to solve these problems, this paper proposes a SWIM cross-domain authentication scheme based on a consistent hashing algorithm on consortium blockchain and designs a blockchain certificate format for SWIM cross-domain authentication. The scheme uses a consistent hash algorithm with virtual nodes in combination with a cluster of authentication centers in the SWIM consortium blockchain architecture to synchronize the user's authentication mapping relationships between authentication domains. The virtual authentication nodes are mapped separately using different services provided by SWIM to guarantee the partitioning of the consistent hash ring on the consortium blockchain. According to the dynamic change of user's authentication requests, the nodes of virtual service authentication can be added and deleted to realize the dynamic load balancing of cross-domain authentication of different services. Security analysis shows that this protocol can resist network attacks such as man-in-the-middle attacks, replay attacks, and Sybil attacks. Experiments show that this scheme can reduce the redundant authentication operations of identity information and solve the problems of traditional cross-domain authentication with single-point collapse, difficulty in expansion, and uneven load. At the same time, it has better security of information storage and can realize the cross-domain authentication requirements of SWIM users with low communication costs and system overhead.

KEYWORDS

System-wide information management (SWIM); consortium blockchain; consistent hash; cross-domain authentication; load balancing

1 Introduction

With the rapid growth of civil aviation air traffic volume and the continuous improvement of the requirements for the civil aviation service level, efficient business interaction between different units has become a key problem in the field of civil aviation. Traditional civil aviation systems communicate in a point-to-point manner, but the information systems used by different institutions are heterogeneous and lack uniform interface standards, and the further development and maintenance of information systems will entail huge costs. The concept of system-wide information management (SWIM) provides an effective to solve the above problems [1,2].

1.1 Motivation

SWIM, as the core of the next-generation air traffic management system, serves to connect various sub-information systems of the existing civil aviation information network, integrating various types of service resources in civil aviation, promoting information interaction and data sharing in the field of civil aviation [3], and enhancing the operational efficiency of civil aviation. The SWIM system involves the authentication of multiple domains, and it is crucial to ensure secure access between different domains. The purpose of studying cross-domain authentication is to design and develop secure and reliable authentication mechanisms to prevent unauthorized access and data leakage. This helps to protect sensitive information and ensure the security of the system, to build trust relationships, to enhance the exchange and sharing of data between domains, thus improving the efficiency and accuracy of the whole system.

1.2 Problem Statement

Due to the service-oriented architecture and distributed deployment of SWIM, civil aviation units face some challenges in interfacing with SWIM to access business resources in other regions. At present, the research and deployment of civil aviation SWIM networks is still in the early stage, and the public key infrastructure (PKI) system has not yet been fully established. When building the SWIM service architecture, we take into account the development of the civil aviation industry and the actual situation of guest distribution, users are mainly differentiated by region, and the information network authentication mechanisms of civil aviation units in different geographic regions are independent of each other. This leads to the problem of duplication of certificate applications and high idleness of certificates. Users mapped by authentication relationships that already existed before civil aviation units accessed SWIM lack reasonable authentication mechanisms to access services and resources in other SWIM authentication domains. In addition, aviation service providers in different regions may need to join SWIM at any time, and the traditional cross-domain authentication system lacks a flexible authentication node expansion strategy. Meanwhile, the shared data in the SWIM system is based on the existing civil aviation network information system and distributed systems, and there are business interaction problems in information consumption and service distribution in different authentication domains, which provide hackers with opportunities to intrude and conduct malicious attacks. This poses a great challenge to the compatibility of SWIM authentication relationships and the security of users during cross-domain access. The traditional "push mode" and "pull mode" have the problems of physical proximity of authentication servers and internal network connection between servers, which are not suitable for distributed deployment of the SWIM system. They also have the problem of receiving incorrect authentication due to "man-in-the-middle attacks" [4]. The traditional authentication scheme based on consistent hashing inevitably has the problem of load balancing, which needs to be solved by changing the load balancing strategy of the authentication service. In order to solve the above problems, the consortium blockchain architecture with a consistent hash structure is introduced in the SWIM certification center. The cross-domain authentication method using consortium blockchain technology can effectively solve the problems of single-point collapse, expansion difficulties, and uneven workload distribution in the authentication center for cross-domain authentication. In order to achieve secure authentication of SWIM data sharing and flexible decision-making for cross-domain authentication, the SWIM authentication center introduces a consistent hash structure and uses the consortium blockchain for data synchronization during the data synchronization phase of the authentication information certification server. The coalition chain has the characteristics of global data consistency, non-tampering, faster transaction speed, and low transaction cost, and retains other characteristics of the blockchain, so it is very suitable for cross-domain authentication and data synchronization in the SWIM environment.

1.3 Contribution

The main contributions of this paper include the following:

- Establish an architecture of the consortium blockchain network for SWIM center nodes, and realize the authentication center nodes in different regions to join the SWIM consortium blockchain network through the consortium blockchain access mechanism.
- A cross-domain authentication scheme for consortium blockchain based on consistent hashing is proposed, which constructs the SWIM regional authentication center hash ring and the service authentication hash ring, respectively, and designs the consortium blockchain certificate format for SWIM authentication.
- A cross-domain authentication protocol for SWIM is designed to realize secure and effective cross-domain access for users in different authentication domains of SWIM.
- A SWIM-oriented coherent hash ring-cutting method is proposed, which exploits the characteristics of the services provided by SWIM in order to achieve load balancing on the service authentication hash ring for user authentication requests.

The rest of the paper is organized as follows: Section 2 describes the work related to the study of efficient and secure cross-domain authentication for SWIM. Section 3 describes the model of SWIM cross-domain authentication, the design of blockchain certificates for cross-domain access services, and the consortium blockchain cross-domain authentication protocol with consistent hashing. Section 4 describes the various stages and application scenarios of the cross-domain authentication scheme implementation. Section 5 discusses the security and performance analysis of the scheme. The conclusion in Section 6 presents limitations, discussion, and future work.

2 Related Work

Many scholars have conducted a series of studies on cross-domain authentication. Two research threads are related to this paper, namely, the application scheme based on consistent hash and the cross-domain authentication solution based on blockchain. The consistent hash algorithm was first proposed by Karger et al. [5] in 1997. The initial goal of the algorithm was to achieve load balancing and dynamic adaptation on dynamically changing distributed systems. Later, consistent hash algorithms were used in several fields, and cross-domain authentication was one of them. Wu et al. [6] proposed the grouped consistent hash cut method, which alternately maps nodes into the consistent hash space in groups according to their computational capabilities. Nakatani [7] used structured allocation of consistent hashes to improve the balancing characteristics of cloud infrastructures for fast response to meet fault tolerance as well as slow updates for node scaling. Thar et al. [8] used coherent hashing to address writing cache decision-making and forwarding mechanisms in content centric-networks. Yao et al. [9] proposed a Web cross-domain authentication optimization scheme using a consistent hash structure to achieve user authentication in different authentication domains by mapping authentication servers and application servers in the same consistent hash space, but the authentication servers in the scheme synchronize their data through internal local area networks (LANs) without considering the problem of coping with the “man-in-the-middle attacks.”

The emergence of blockchain technology [10] and smart contracts [11] simplifies the cross-domain authentication process. For cross-domain authentication between trust domains using public key infrastructure and trust domains using identity-based cryptography, Ma et al. [12] set up a blockchain domain proxy server in the trust domain of identity-based cryptography to participate in key generation. Chen et al. [13] designed a trust model for blockchain certificate authority centers to improve the efficiency of cross-domain authentication while reducing the number of public key algorithms for signing and verifying. Xue et al. [14] proposed a blockchain-based full cross-domain authentication scheme to address the challenge of incomplete cross-domain authentication, which allows users to use different cryptographic settings without relying on a trusted third party. Wang et al. [15] used consortium blockchain technology to build a decentralized network with root certificate authorities as authentication nodes, which omit the overhead of key encryption and decryption in the cross-domain authentication process. Zhang et al. [16] designed a cross-domain authentication scheme based on a multi-layer blockchain, using local blockchain for distributed node management in IoT application scenarios and public blockchain for cross-chain authentication between blockchains, which improves the efficiency of cross-domain authentication. In order to address the complexity and technical implementation difficulties of scaling systems on the blockchain, Ali et al. [17] put forward the hierarchical structure of the local and global intelligent contract and used the proof of authenticity and integrity mechanism to retrieve and find the user or IoT device platform hash stored on the local blockchain network during cross-domain identity authentication. This scheme has the environment of a large number of concurrent requests to generate a high throughput capability with less computing cost. Li et al. [18] designed a cross-domain authentication and key agreement protocol based on the smart contract of blockchain, but it is not necessarily applicable to authentication in civil aviation. Some difficulties in the cross-domain access between heterogeneous civil aviation information networks are encountered in SWIM. SWIM has a distributed deployment nature, and a certain area or unit needs to gradually access SWIM. Therefore, the expansion of the authentication domain and the addition or deletion of authentication relationships cannot have a great impact on the overall authentication architecture of SWIM.

The SWIM operation relies on a service-oriented architecture, and consumers have large differences in the number of access requests for services of different types and from different regions. A network bottleneck or single point of failure may exist. Therefore, this paper presents a cross-domain authentication scheme for the SWIM consortium blockchain based on the consistent hash. First, the scheme sets up a regional certification center hash ring and a service authentication hash ring. This scheme reduces the difficulty of SWIM public authentication domain extension. The consensus of the authentication mapping relationship between different SWIM authentication domains and the cross-domain authentication of user access services is realized. Second, the scheme also combines the Security Assertion Markup Language (SAML) assertion token mechanism in the SWIM system to reduce the redundant authentication operations of the identity information. Finally, the scheme is used to design a consistent hash ring-cutting method for the SWIM service to achieve the load balancing of authentication requests. The scheme can resist the "man-in-the-middle attacks" [19], the "replay attacks" [20], and the "Sybil attacks" [21]. Compared with the traditional consistent hash cross-domain authentication scheme, the proposed scheme improves the security of user authentication mapping synchronization and realizes the load balancing of different service authentication requests. Therefore, this scheme is more suitable for the SWIM environment with many concurrent requests.

3 Design of Cross-Domain Access Scheme

3.1 Cross-Domain Access System Model

To achieve the goal of SWIM cross-domain authentication, based on the scheme by Yao et al. [9], this study designed a cross-domain authentication model of the SWIM consortium blockchain based on consistent hash, as shown in Fig. 1. It consists of the consortium blockchain network, the regional CA hash space, the service authentication hash space, the domain, the domain CA server, the domain proxy authentication server, the application server, and the user. Table 1 shows the notations involved in the proposed scheme.

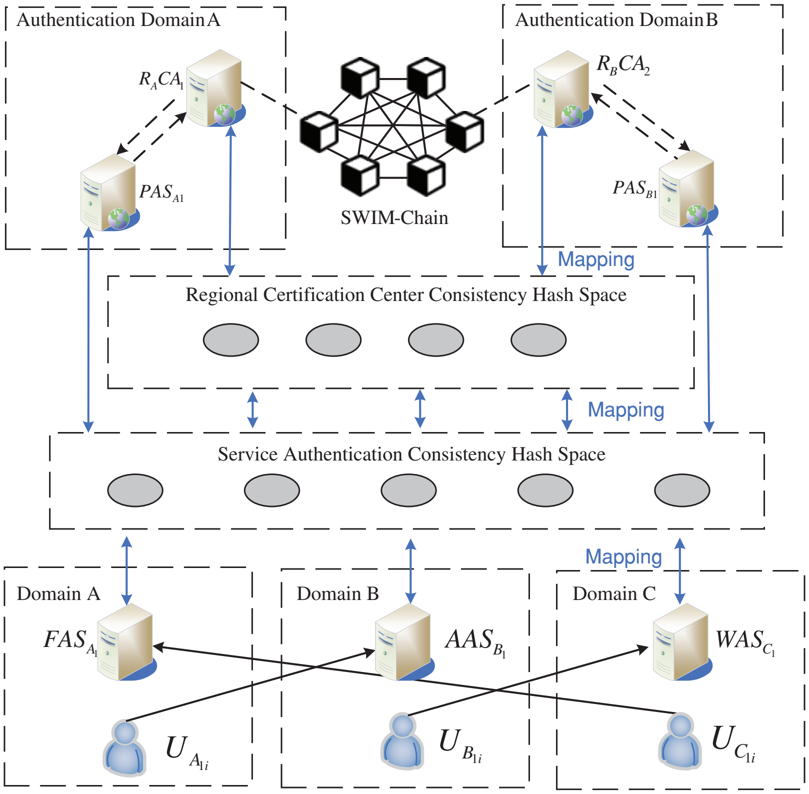

Figure 1: Cross-domain authentication model of SWIM consortium blockchain based on consistent hash

Table 1: Some notations used blow

Notations	Description
CHS_{RPAS}	Region authentication consistent hash space
CHS_{SC}	Service authentication consistency hash space
FAS_{A1}	Domain A flying application server
AAS_{B1}	Domain B aviation application servers
WAS_{C1}	Domain C meteorological application server
R_ACA_1	Authentication center server in authentication domain A

(Continued)

Table 1 (continued)

Notations	Description
PAS_{A1}	Authenticating domain proxy authentication servers in domain A
U_{A1i}, U_{B1i}, U_{C1i}	Domain user
TID	Thread identifier

Domain: A domain is an area of mutual trust between users in the same organization, such as an airline or an air traffic control authority that manages a region. The authentication architecture and relationship of different domains are independent.

Application Server: The SWIM area provides different services according to different functions and the civil aviation network. The service types can be divided into flight, aeronautical, and weather information categories, such as the air traffic control departments, to provide aviation surveillance data and meteorological services. The airport provides ground information services. In Fig. 1, the flight application server in Domain A is denoted as FAS_{A1}.

CA node: The CA node mainly handles query authentication requests from the intra-domain proxy authentication server and the inter-domain CA node, records user registration and cross-domain authentication applications in the form of transactions on the chain, and participates in the consensus of the transactions of CA nodes in other domains. The CA server in Authentication Domain A in Fig. 1 is denoted as R_ACA_1.

Domain proxy authentication server: The domain proxy authentication server mainly deals with user authentication requests from the same or different domains. This server sets up three types of service authentication service nodes for flight, aeronautical, and weather information application server authentication requests and increases or decreases the number of servers according to the authentication load balance. The domain proxy authentication server in Authentication Domain A in Fig. 1 is denoted as PAS_{A1}.

User: In SWIM, a user can be both a service provider and consumer and can be described as an entity that owns a specific resource in one domain while possibly needing to access other services in other domains. In Fig. 1, the users in Domain A are denoted as U_{Ai}.

Region authentication consistent hash space CHS_{RPAS}: The hash function is designed to calculate the hash space of the virtual authentication hash ring, and the different regional authentication center server nodes are mapped into the authentication hash ring through the hash operation. The virtual service nodes in the lower service authentication hash space map the cross-domain authentication requests object to the CHS_{RPAS} and find the authentication center server node for processing.

Service authentication consistency hash space CHS_{SC}: The hash function is designed to calculate the hash space of the virtual service hash ring, and the domain proxy authentication node maps three kinds of service authentication virtual nodes to the CHS_{SC} to take charge of the three types of service authentication in the domain. Meanwhile, the corresponding service requests from users inside and outside the domain received by the application server node are mapped into the service authentication hash ring by the hash operation, and the service authentication virtual node is searched for processing.

Consortium blockchain network: This network is composed of SWIM regional certification center nodes. The certification center servers in different domains join the SWIM-Chain after receiving permission and can interact and consensus authentication information through the consortium blockchain network. The consortium blockchain network in the model designed in this paper is built using the consortium blockchain platform Hyperledger Fabric [22].

3.2 Blockchain Certificate Design

A certificate is designed for the SWIM-Chain cross-domain access service application scenario. The certificate contains the blockchain certificate used by the CA node and the user and the domain proxy temporary certificate used by the domain proxy server. The blockchain certificate is generated by the initial registration domain of the user and recorded into the blockchain ledger after a consensus is reached by all the nodes of the consortium blockchain. The certificate can be used as a trusted credential for users to access other domain services. The domain proxy temporary certificate is registered and generated by the domain certification authority for identity verification when interacting with other domain proxy authentication servers. The blockchain certificate and domain proxy temporary certificate for SWIM is shown in Fig. 2.

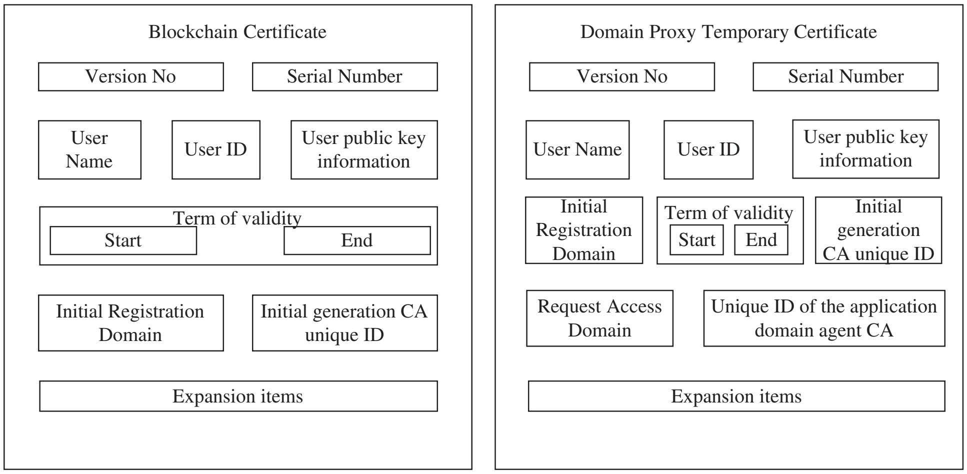
Blockchain Certificate
Version No
Serial Number
User Name
User ID
User public key information
Term of validity
Start
End
Initial Registration Domain
Initial generation CA unique ID
Expansion items
Domain Proxy Temporary Certificate
Version No
Serial Number
User Name
User ID
User public key information
Initial Registration Domain
Term of validity
Start
End
Initial generation CA unique ID
Request Access Domain
Unique ID of the application domain agent CA
Expansion items

Figure 2: Blockchain certificate and domain proxy temporary certificate for SWIM

The blockchain certificate designed in this paper eliminates the function of certificate revocation checking. Given that the data stored in the blockchain database cannot be directly modified, the traditional X.509 certificate revocation service cannot be applied to the blockchain scenario. According to the literature [23], this paper defines the interface of the certificate store on the chain as a post (action, Hash_X (Cert)), where action represents the current state of the certificate, which can be either issued or revoked. The Online Certificate Status Protocol and the Certificate Revocation List can be replaced by changing the certificate validity by changing the status parameter action. The interface to query a certificate is defined as a query (Hash_X (Cert)), which returns the certificate details and the current status of the certificate.

3.3 Cross-Domain Authentication Protocol for Consortium Blockchain Based on Consistent Hash

According to the above model of cross-domain authentication and the application scenario of consistent hashing in the SWIM-Chain, this paper presents a cross-domain authentication protocol for the consortium blockchain based on consistent hashing. The following are assumed: the authentication domain joined by the consortium blockchain access mechanism is credible, and the certificate information of the root CA of each authentication domain has been stored in the block of the blockchain before the cross-domain authentication protocol starts, as shown in Fig. 3.

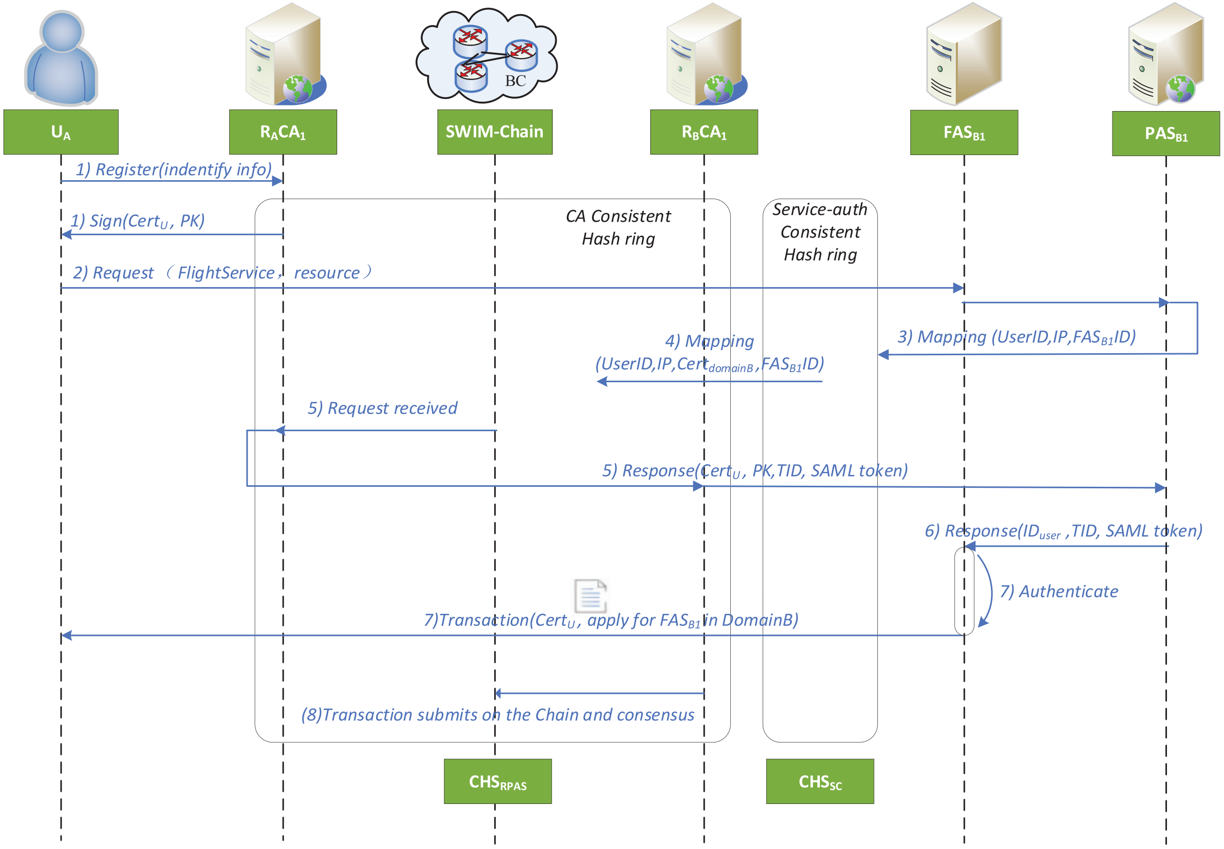
BC
U_A
R_ACA_1
SWIM-Chain
R_BCA_1
FAS_{B1}
PAS_{B1}
1) Register(indentify info)
1) Sign($Cert_U$, PK)
CA Consistent Hash ring
Service-auth Consistent Hash ring
2) Request （FlightService， resource）
3) Mapping (UserID,IP,FAS_{B1}ID)
4) Mapping (UserID,IP,$Cert_{domainB}$,FAS_{B1}ID)
5) Request received
5) Response($Cert_U$, PK,TID, SAML token)
6) Response(ID_{user} ,TID, SAML token)
7) Authenticate
7)Transaction($Cert_U$, apply for FAS_{B1} in DomainB)
(8)Transaction submits on the Chain and consensus
CHS_{RPAS}
CHS_{SC}

Figure 3: Flow chart of cross-domain authentication protocol

1) R_ACA_1→U_A

Taking the cross-domain authentication between Domains A and B as an example, Domain A User U_A first needs to register with the certification center of Domain A and obtain the unique blockchain certificate (CertU_A) issued by the certification center.

2) U_A→FAS_{B1}

User U_A in Domain A requests to access the data resource of flight application server FAS_{B1} in Domain B.

3) FAS_{B1}→PAS_{B1}→CHS_{SC}: Mapping{UserID, IP, FAS_{B1}ID}

Domain B's flight application server FAS_{B1} receives the access requests of Domain A User U_{Ae} and maps the UserID, the IP address used to access the service, the ID of FAS_{B1}, and other information to the service authentication consistency hash space (CHS_{SC}) by domain proxy authentication PAS_{B1}. Meanwhile, the first service authentication virtual node responsible for the same service is searched clockwise to handle the authentication request. The service authentication node queries the certificate information of the user in the local database and parses the certificate according to the information provided by User U_A to check the validity period of the certificate. If the validity test is passed, then the successful authentication information and the Thread Identifier (TID) are returned to FAS_{B1}. If the certificate information of the user is not queried locally, the domain proxy authentication server PAS_{B1} will carry out the next mapping transfer.

4) $CHS_{SC}\rightarrow CHS_{RPAS}$: Mapping{UserID, IP, $Cert_{domainB}$, FAS_{B1} ID}

If the service authentication hash space (CHS_{SC}) receives a user application that is not archived on the virtual service authentication node to which the SWIM node is mapped, the virtual service authentication node should send the request for certification relationship (RCR), which is mapped to the domain proxy authentication consistency hash space (CHS_{RPAS}), and the first server is searched in the clockwise direction. If there is no authentication mapping relationship information for the user, skip this point and continue searching for the next node. If relevant user authentication information exists, then the node is selected as the RCR service node.

5) $R_ACA_1\rightarrow R_BCA_1\rightarrow PAS_{B1}$: Response{CertUA, PKuserA, TID, SAML token}

After R_AC_{A1} receives the request, the A-domain CA node returns the unique blockchain certificate (CertUA), the public key (PKUA), the TID, and the SAML token of User U_A to the sender of the requests and writes the authentication relationship into the block transaction, which is broadcast to other nodes according to the Practical Byzantine Fault Tolerance (PBFT) algorithm during the node consensus. The next time the user accesses other service authentication nodes, it can return the information of the first node queried clockwise on the service authentication node mapped to the regional authentication hash ring.

6) $PAS_{B1}\rightarrow FAS_{B1}$: Response{IDuserA, TID, SAML token}

The domain proxy authentication (PAS_{B1}) returns the user ID, the TID, and the SAML token that passed the cross-domain authentication to the flight application server (FAS_{B1}), and the token is parsed and verified.

7) $FAS_{B1}\rightarrow U_A$

If the user ID exists and the security assertion is valid, then the required service is returned. Finally, the flight application data resources are provided to User U_A to realize the cross-domain authentication between the user in Domain A and the application server in Domain B.

8) Re-authentication

When Domain A User U_A applies for flight application services in Domain B again, the Domain B application server hashes the user's unique identity and application address into the service authentication hash ring. The unique blockchain certificate ($CertU_A$) of User U_A with the unique identity and signature of the Domain B certification center on the chain has been the consensus. Therefore, the Domain B certification center server directly performs the hash operation according to the unique identity of the user, querying the blockchain status database and verifying the validity of the certificate to complete the authentication.

4 Implementation and Application of SWIM Cross-Domain Authentication Scheme

4.1 Implementation

4.1.1 System Initialization

During system initialization, the public and private keys of all the entities in the aeronautical information domain are generated. The SWIM-Chain blockchain network is established, and the Chain code is deployed on the nodes. The blockchain certificate information of the entities in the aeronautical information domain is stored in the blockchain network, and this phase is performed once.

1) Generate the public-private key pair

All entities in all aeronautical information domains accessing SWIM, including the CA server, the domain proxy authentication server, the application server, and all users, initialize their public and private key pairs according to the encryption mechanism adopted in the local domain.

2) Build a SWIM-Chain blockchain network and deploy chain code

When the certification centers of Domains A and B are licensed through the consortium blockchain access mechanism, R_ACA and R_BCA join the SWIM-Chain network, and the chain code is deployed to all the nodes. The designed chain code includes three functions, namely, registration, query, and verification, and each function can be called with its function identifier and proper input parameters.

- NodeRegister $\left(ID_{R_XCA}, pk_{R_XCA}, Hash_X\left(ID_{R_XCA}, pk_{R_XCA}\right), Hash_X, sign_X\right)$:

This chain code function handles the registration requests of the authentication server nodes pre-joining the blockchain network and stores the encryption settings information pre-used by the authentication server, triggered by the chain code identifier NodeRegister and related input parameters. ID_{R_XCA} is the identity of the authentication server R_XCA, pk_{R_XCA} is the public key pre-generated by the authentication server R_XCA, and ID_{R_XCA} is the public key of the authentication server R_XCA. $Hash_X\left(ID_{R_XCA}, pk_{R_XCA}\right)$ is the hash value of ID_{R_XCA} and pk_{R_XCA}, and $Hash_X$ and $sign_X$ are the respective hash and signature algorithms adopted by the authentication server. After calling the chain code, if the input parameter information meets the registration conditions, it is stored on the blockchain, and the blockchain certificate $Cert_{R_XCA}$ is returned to the authentication server node.

- UserRegister $\left(ID_{UXi}, pk_{UXi}, Hash_X\left(ID_{UXi}, pk_{UXi}\right), Hash_X\left(ID_{R_XCA}, pk_{R_XCA}\right)\right)$:

This chain code function handles the registration requests for the user's identity information from the domain. This function can be triggered by the chain code identifier UserRegister and associated input parameters, where ID_{UXi} and pk_{UXi} represent the user's unique identity and pre-generated public key. $Hash_X\left(ID_{UXi}, pk_{UXi}\right)$ and $Hash_X\left(ID_{R_XCA}, pk_{R_XCA}\right)$ denote the respective hash values of $\left(ID_{UXi}, pk_{UXi}\right)$ and $\left(ID_{R_XCA}, pk_{R_XCA}\right)$ computed by the hash algorithm for Domain X. After calling the chain code, if the input parameter information meets the registration conditions, it is stored on the blockchain, and blockchain certificate $Cert_{UXi}$ is returned to the user node.

- VerifyCertInfo $\left(ID_{UXi}, pk_{UXi}, Cert_{UXi}, Hash_X\left(ID_{UXi}, pk_{UXi}\right), Cert_{UXi}, sign_X\left(sk_{UXi}, N\right), N\right)$:

This chain code function is used to verify the cross-domain access authentication requests from users in other domains to Domain X. This function can be triggered by chain code identifier VerifyCertInfo and related input parameters, where ID_{UXi} and pk_{UXi} represent the respective unique identity and pre-generated public key of user U_{X_i}, and $Cert_{UXi}$ is the blockchain certificate generated by the initial registration domain after the user is registered. $Cert_{UXi}$, $Hash_X\left(ID_{UXi}, pk_{UXi}\right)$, $Cert_{UXi}$ is the hash value of the relevant identity information of user U_{Xi}, and $sign_X\left(sk_{UXi}, N\right)$ is the signature value generated by signature algorithm $sign_X$ by user private key sk_{UXi} and random number N. If the user passes the cross-domain authentication, the SAML assertion token is returned to the user node for the user node to request service from application server AS.

- CrossdomainAuthInfoShare $\left(ID_{UXi}, pk_{UXi}, Hash_X\left(ID_{UXi}, pk_{UXi}, Cert_{UXi}\right), sign_Y\left(Cert_{UXi}\right)\right)$:

This chain code function is used to locally store and share user identity information that has been authenticated across domains in Domain Y. The authentication relationship is written into the block transaction and broadcasted to other nodes when the node reaches consensus so that other nodes can store it in the on-chain state database. This function can be triggered by chain code identifier CrossdomainAuthInfoShare and associated input parameters, where ID_{UXi} and pk_{UXi}, respectively, represent the unique identity and pre-generated public key of user U_{X_i}. $Hash_X\left(ID_{UXi}, pk_{UXi}, Cert_{UXi}\right)$ is the hash value generated by the initially generated domain hash algorithm for the relevant identity information of User U_{X_i}, and $sign_Y\,(Cert_{UXi})$ is the signature value generated by signature algorithm $sign_Y$ for the Domain Y information that passes the user's cross-domain authentication for the user's blockchain certificate.

3) Store the blockchain certificate information of entities in the aeronautical information domain in the SWIM-Chain

When the authentication server node or user registers to join the blockchain, the entity identity information and the generated blockchain certificate should be stored in the blockchain, and the NodeRegister and UserRegister chain codes are called to store the identity information in the form of key-value pair into the blockchain state database. For example, the corresponding and user authentication server node key-value pair data structure is $(Hash_X\left(ID_{R_XCA}, pk_{R_XCA}\right)|Cert_{R_XCA},\ Hash_X, sign_X)$ and $(Hash_X\left(ID_{U_{X_i}}, pk_{U_{X_i}}\right)|Cert_{U_{X_i}})$, respectively.

4.1.2 Authentication Load Balancing in Consistent Hash Paces

Consistent hash space CHS_{SC} performs load balancing of service authentication by adjusting the coverage of the service authentication nodes in the hash space. A CA node maps three kinds of service authentication nodes, providing different numbers of the three kinds of service authentication nodes according to the proportion of the three kinds of services provided in the area, and distributes the nodes to the consistent hash ring. The region division dynamically changes the number of the secondary virtual nodes of various service authentication nodes according to the number of services provided and the number of applications for services, increases their distribution range on the hash ring, and speeds the time of user cross-domain access authentication.

Considering the relatively low comprehensive load in the early stage of system operation, the server load situation is not considered in the first *m* cycles T. After *m* periods T, the number of three types of service provision and the number of service requests per unit time are calculated to dynamically plan the service virtual nodes. The number of a certain type of service virtual nodes introduced by each CA node is $k\log|SN|$, where *k* is a constant, and *SN* is the total number of virtual service nodes. If the number of services provided by the service authentication node is A_i, the number of times the application service is received in unit time T is a_i, and the number of nodes is *N*. Then, the number of consistent hash virtual nodes that must be allocated by the service authentication node is $\frac{A_i a_i}{\sum a_i}k\,|SN|$, and the total number of consistent hash virtual nodes that must be allocated in the system is $\sum_{i=0}^{N}\frac{A_i a_i}{\sum a_i}k\,|SN|$.

When the service application times of different service authentication nodes in unit time are very different, the center of gravity of the total number of virtual nodes set by the system will shift to the service authentication nodes that apply for more service times. Given that the lookup process of the consistent hash algorithm is based on the traversal of the tree structure, when the number of virtual nodes increases, the distribution range of the service authentication nodes in the consistent hash space increases, and the lookup time decreases. Although the performance improvement of the multi-request service authentication node requires the sacrifice of a small part of the performance of other nodes, the imbalance of task distribution and the performance unsaturation of the authentication server are improved, thereby improving the overall performance of the system.

4.2 Application Scenario of the Consistent Hashing in SWIM-Chain

The consistent hash algorithm is applied in this scheme to reasonably select the authentication service nodes and distribute the authentication service objects. The storage space of the authentication processing object corresponding to the authentication requests object is abstracted as a circular closed hash ring with 2 ∧ 32 points. The unique identity of the object is mapped into the hash ring through the hash algorithm. Then, the first authentication server node is searched clockwise according to the position mapped by the authentication requests, and the request is processed. In this paper, the Fowler-Noll-Vo (FNV) algorithm is used as the hash algorithm in the above steps. The application scenario of the consistent hashing algorithm in the proposed scheme is shown in Fig. 4.

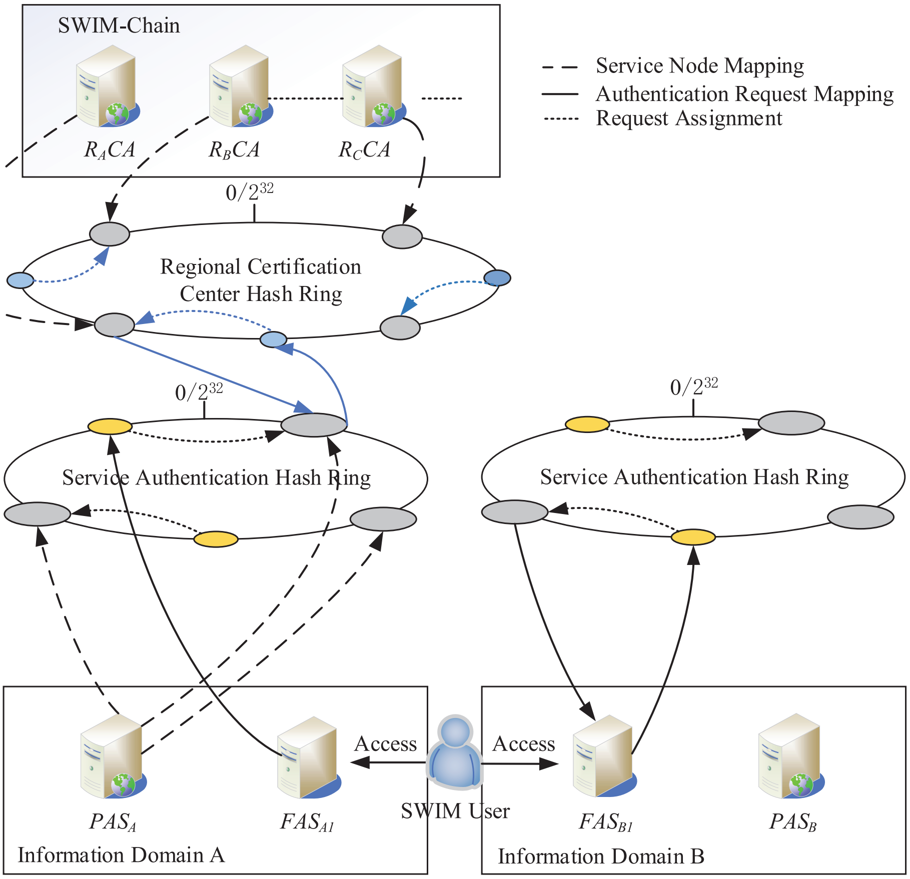

Figure 4: Application scenario of the consistent hashing algorithm in SWIM cross-domain authentication

Each information domain has an independent service authentication consistency hash space (CHS_{SC}). CHS_{SC} uses the IP address, port number, and service type number of domain proxy authentication server PAS_{A1} as keys to map three kinds of service authentication virtual nodes to realize the processing of three types of service authentication in the domain. Different information domains share a regional authenticated consistent hash space, CHS_{RPAS}. CHS_{RPAS} uses the IP address and port number of the server nodes in different domains as keys and maps real service nodes through the FNV algorithm to process cross-domain authentication requests, and the algorithm flow is shown in Algorithm 1.

Algorithm 1: Consistent Hashing Algorithm in SWIM-Chain

```
Input: Request: user request, IP: user IP address, Type: request service type
Output: SAMLToken: SAML Assertion Token, Au_Result: Authentication Result
For i to h
  Key = (Request, IP, Type)
  A = FNV (Key)
  If $PAS_A$ user identity information A in {A1, A2, ..., A3}
    return SAMLToken to the user
End For
Else
    Mapping RCR to $CHS_{RPAS}$
    clockwise direction
      For 1-n $CHS_{RPAS}$
        If the $R_A CA_1$ is found
          Query the authentication mapping relationship information for this user
          If There is user authentication information
            Return Cert and Sign to request sender
            If the sender of the request passes the authentication
               return SAMLToken to user
            End If
            Write the authentication relationship to the block transaction
            Broadcast to other nodes using the PBFT algorithm during node consensus
          End If
        End If
      End For
End If
return Au_Result
```

When the domain application server node receives a service request from a user outside the domain, it will map the user's IP address and the type of the application that serves as the key through the FNV algorithm into the service authentication hash ring and find the first service authentication node responsible for the same service to submit the authentication requests in the clockwise direction. If the domain proxy authentication server locally stores the identity information of the requesting user, the SAML assertion token will be returned to the user. If the domain proxy authentication server does not find the identity information of the applying user locally, the RCR should be mapped to the hash ring of the regional certification center through the service authentication virtual node, and the first server should be searched in the clockwise direction. If no authentication mapping information of the user exists, the node is skipped, and the next node is searched. If an application for user authentication information exists, the blockchain certificate and signature should be returned to the request sender, and the request sender verifies the signature and blockchain certificate to return the SAML assertion token to the user while writing the authentication relationship into the block transaction. According to the PBFT algorithm [24], the authentication relationship broadcasts to other nodes when the node reaches consensus. The next time the user visits other service authentication nodes, the first node queried clockwise on the service authentication node mapped to the regional authentication hash ring can return the correct information.

5 Security and Performance Analysis

5.1 Security Analysis

By building the consortium blockchain platform Hyperledger Fabric, this paper analyzes the ability of the scheme to resist man-in-the-middle, replay, Sybil, and distributed denial of service attacks under specific attacks patterns and scenarios and uses two-way entity authentication and distributed trust based on the consortium blockchain. Thus, the security of the system is improved.

A) Resist man-in-the-middle and replay attacks. Because the user is mapped to a hash ring, required information, such as the port and ID, is different, making the map to hash the location of the ring different, leading to accepting the proxy authentication server being different also. If a user applies for cross-domain authentication twice in a row and is accepted by two different domain proxy authentication servers, the second domain proxy authentication server needs to verify the identity information of the applying user, such as the certificate. During the mapping of the authentication requests to the domain proxy authentication, the position of the node on the consistent hash ring is random. In other words, the proxy server handling the cross-domain requests may not be in the target domain. However, according to the consensus of Swim-Chain authentication information and the participation of all domains in the process of cross-domain authentication, it can ensure that the user's cross-domain authentication request is processed securely and efficiently.

B) Resist sybil attacks. Among the attacks on the blockchain system, the Sybil attack is an attack on identity management rules, which uses malicious nodes in the system to request many irregular transactions to influence the direction of the blockchain system and even control the system's accounting rights. Given that the experimental scenario of this scheme is inside the air traffic control system, the selected nodes have a high degree of credibility compared with the nodes of the public chain system. In addition, the blockchain certificate is designed for the cross-domain authentication model to standardize the registration management of the identity of the certification center and user nodes, reducing the proportion of malicious nodes in the system.

C) No trusted third party is required. Unlike traditional "non-isolating" civil aviation information system networks, the architecture of a consortium blockchain adopts distributed trust and does not rely on a third-party certification body, using the point-to-point communication mode between nodes, node participation by members of the distributed deployment of SWIM cross-domain authentication process, each node will store user cross-domain authentication key data. In this approach, the system paralysis caused by malicious attacks on a single certification authority is avoided, and the security and scalability of the system are guaranteed.

D) Mutual entity authentication. In each authentication domain, the authentication between the user and the authentication server is realized by the original authentication method in the domain. Under the architecture of the multi-authentication domain consortium blockchain, the user applies for a blockchain certificate from the domain proxy authentication server, and the target server queries and applies for the user authentication mapping relationship and trust credential to confirm the trust relationship and realizes the authentication between the user and the target domain server.

E) Resist distributed denial of service attacks. The cross-domain authentication model of the SWIM consortium blockchain designed in this paper has a perfect access mechanism for authentication center nodes in different authentication domains and user nodes in the domain, standardizing the credibility of users. Meanwhile, the scheme uses the decentralized architecture of blockchain, which has the characteristics of decentralization, redundancy, fault tolerance, dynamic regulation, and vulnerability without a single point of failure of the distributed system. Therefore, the proposed authentication protocol can effectively resist distributed denial of service attacks.

5.2 Performance Analysis

SWIM-Chain is minimized to three organizations, including six Peer nodes, three Orderer nodes, and one CA node, and the network topology is shown in Fig. 5. One Peer node acts as a commit node, Committer, and one endorsement node, Endorser. The Committer is responsible for checking the block transaction data sorted by the Orderer node, selecting legitimate transactions to execute and write to storage, and finally sending the block information to other endorsement nodes for consensus. Check whether the block transaction information published by the Committer node is legal, and endorse and sign if it is a legal transaction. Initialization of the SWIM-Chain system refers to the process of building the consortium blockchain network and installing the instantiated chain code, which is the responsibility of the SWIM platform management. The process covers modifying configuration files, starting Docker containers, starting functional nodes, creating channels, nodes joining channels, and installing and instantiating chain codes. In our study, this paper used a server cluster scenario with microservices architecture, where the server is equipped with an Intel CPU 10700 @ 2.40 GHz processor and 128 GB of RAM and is running a CentOS 7 (64-bit) operating system. This paper built the blockchain network based on Fabric version 1.4.1 and used the Go language for Fabric development. In addition, this paper utilized the open-source container engine Docker and the Beego framework to provide interface requests for the SDK, respectively.

5.2.1 Cost of Computation

Analyzing the computational cost of the scheme requires an analysis of the computational overhead of the scheme and a comparison of this scheme with existing schemes. The computational overhead is shown in Table 2. In particular, to ensure the smooth implementation of the scheme without loss of generality and balance the complexity of different schemes, this paper sets the number of member nodes of this scheme and the other schemes to 2. Table 2 shows the comparison of the computational overhead of the four schemes, whose unit is the number of operations, where the two columns of encryption and decryption, signature, and signature verification calculate the sum of the number of distributions.

In terms of encryption/decryption, both the literature [25] scheme and the literature [26] scheme do not involve encryption/decryption operations, while the scheme in this paper involves two encryption/decryption operations. In terms of signature/verify signature, the literature [25] scheme involves twelve signature/verify signature operations, the literature [26] scheme involves six signature/verify signature operations, while the scheme in this paper involves two signature/check-signature operations. In terms of hash operations, the literature [25] scheme involves four hash operations, the literature [26] scheme involves one hash operation, while the scheme in this paper involves two hash operations. The computational overhead of this paper’s scheme in encryption/decryption and signature/verifying the signature is higher than that of the literature [25] scheme and the literature [26] scheme, but it is comparable to the literature [25] scheme and superior to the literature [26] scheme in terms of hash operations.

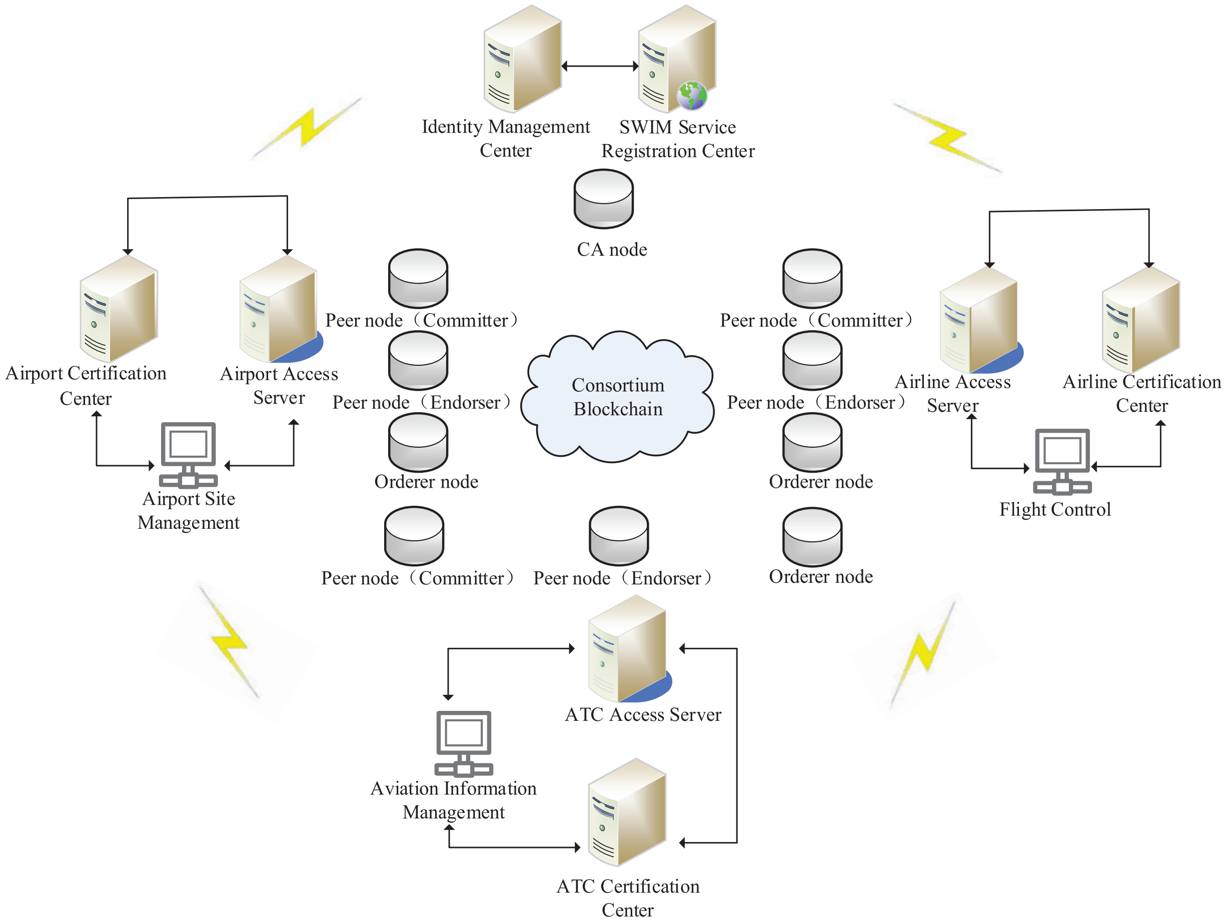

Figure 5: Topology diagram of the SWIM experimental platform

Table 2: Computing overhead comparison

Scheme	Encryption/decrypt	Signature/verify signature	Hashing operation
Literature [25] scheme	0	12	4
Literature [26] scheme	0	6	1
Literature scheme	2	2	2

5.2.2 Efficiency Analysis of Consortium Blockchain Network

In terms of computational efficiency, this paper implements a SWIM-Chain cross-domain authentication scheme prototype to evaluate its performance. In actual simulation experiments, to reflect the operation of the chain code well, 30 experiments were conducted, and each set in each experimental group 500 times of simulation, the nodes register chain code, user registration, chain code, certificate verification chain code, and cross-domain authentication identity Shared chain code running time analysis, this paper put the 20 experiments run time-consuming for statistical data. The experimental results are shown in Fig. 6.

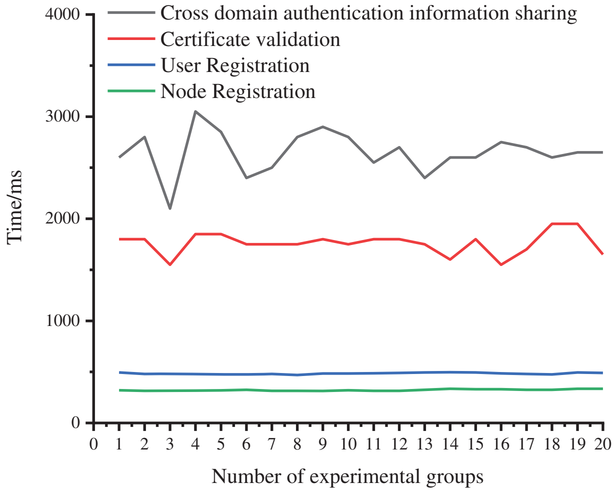

Figure 6: Time required for chain code operation

For the data shown in Fig. 6, conclusions can be drawn through statistics and calculations. The average running time of the node registration chain code is 263.6 ms, the average running time of the user registration chain code is 397.4 ms, the average running time of the certificate verification chain code is 1871.4 ms, and the average running time of the cross-domain authentication information sharing chain code is 69.8 ms. The main reason for the different chain code time consumptions between the experimental groups is the different communication conditions of the peer-to-peer blockchain network, but the overall average time consumption is within the ideal range, so the chain code designed in this scheme is feasible.

This paper obtains the average time consumption of the transaction information consensus process in the cross-domain authentication system through 1000 simulation experiments to reflect its operational efficiency. The consensus process is divided into the Request, Pre-Prepare, Prepare, Commit, and Reply phases, and the average time consumption of each stage of the consensus process is shown in Table 3.

Table 3: Average time spent in each stage of the cross-domain authentication information consensus process

Cross-domain authentication information consensus process	Average time consumption (ms)
Request phase (Request)	53.6
Pre-prepare phase (Pre-Prepare)	186.4
Prepare phase (Prepare)	849.3
Commit phase (Commit)	672.3
Reply phase (Reply)	108.2

In the designed authentication protocol, the PBFT algorithm must be used to reach a consensus for the cross-domain authentication sharing information to ensure that the civil aviation agency nodes reach a consistent confirmation of the user cross-domain authentication status in the distributed environment. In the process of the user's first identity authentication in the SWIM system, two encryptions and decryptions, two signature verifications, and two hash operations are needed, which take 3.241 s in total.

After successfully authenticating the user identity for the first time, the domain proxy authentication server in the SWIM blockchain network adds the authentication credentials and the SAML assertion tokens to the on-blockchain state database in the form of transactions. If the user accesses the services in the domain of other member nodes of the consortium blockchain, the proxy authentication server of the domain can quickly authenticate the user's identity through the deployed certificate verification chain code, greatly reducing the required time.

5.2.3 Load Balancing Efficiency Analysis of Cross-Domain Authentication

This paper evaluated the load balancing efficiency of the cross-domain authentication system by combining the characteristics of consistent hash. The number of consistent hash virtual nodes is increased exponentially. The number of nodes is from 10 to 5000, and the standard deviation of the authentication request allocation and the average response time of authentication are calculated for different numbers of virtual nodes and compared with the findings in the literature [9]. The comparison results are shown in Fig. 7.

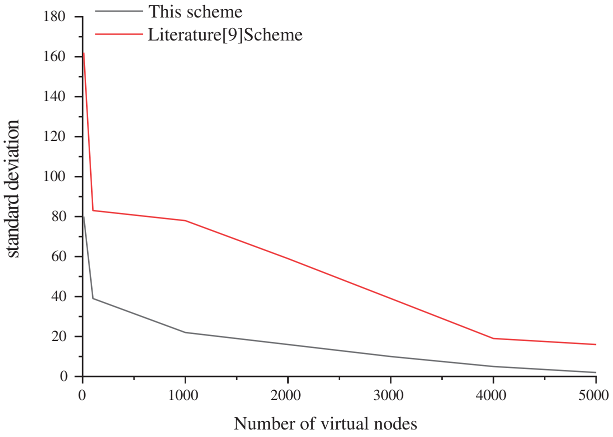

Figure 7: Standard deviation of authentication request allocation

The standard deviation of authentication request assignment decreases substantially with the increase in the number of virtual nodes. The location of nodes on the authentication requests mapping to the domain proxy authentication consistency hash ring is random, and there is an uneven distribution of service nodes in the authentication hash ring. With the increase in the number of virtual nodes, three types of service authentication nodes are distributed to the consistent hash ring according to the proportion of services provided in the area. At this time, the user's authentication request is more likely to be assigned to the correct authentication server, the authentication requests allocation ratio is more average, and the standard deviation of authentication requests allocation is smaller.

When the number of virtual nodes increases, the standard deviation of authentication allocation becomes smaller, the proportion of authentication requests of the three types of services is more reasonable, and the user's authentication requests are more likely to be allocated to the correct authentication server. Thus the average response time of authentication is significantly reduced, as shown in Fig. 8.

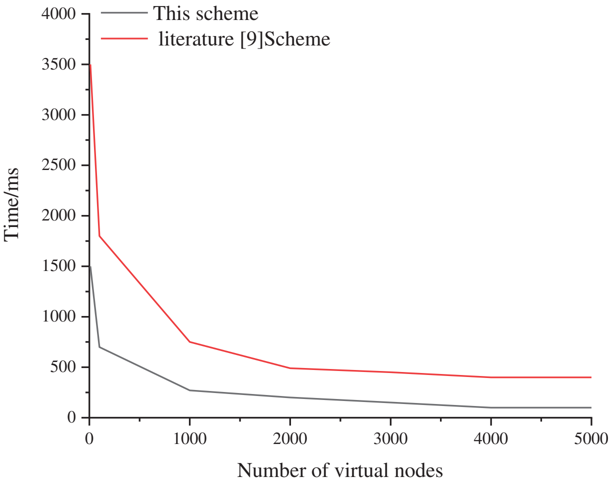

Figure 8: Certification average response time

In this scheme, when the number of virtual nodes increases to 2000–3000, the allocation standard deviation and the average response time of authentication tend to be stable. Theoretically, the higher the number of virtual nodes is, the more balanced the allocation of authentication requests is considering the server performance and the time and space complexity of the consistent hash algorithm. The total number of virtual nodes can be appropriately set as 2000–3000. Therefore, the consistent hash algorithm can be used to realize the load balancing of cross-domain authentication in the SWIM consortium blockchain system.

6 Discussion and Conclusion

6.1 Discussion

The blockchain-based cross-domain authentication system designed in this article meets the requirements of SWIM, takes into account the system characteristics of the SWIM system and civil aviation information system, and adopts blockchain technology to design a shared data authentication system. However, factors such as the identity of civil aviation institutions and users and the level of protection of classified information and information systems can be further optimized for fine-grained partitioning of blockchain nodes in practical situations. This can better meet the needs of different civil aviation institutions and improve the efficiency and flexibility of the system. These issues are worth studying, especially in the field of civil aviation. We will further refine the division of blockchain nodes, taking into account the scale, characteristics, and needs of actual civil aviation institutions, in order to improve the adaptability and efficiency of the system. The data consensus module in the SWIM Chain cross-domain authentication model designed in this article uses the consortium blockchain PBFT included in the Hyperledger Fabric. Currently, the cross-domain authentication results have a high average runtime for information sharing among member nodes in other SWIM Chain. This is because the data synchronization process of PBFT itself is relatively complex, and the method of selecting the master node is relatively cumbersome, which is also its own limitation. The participating nodes in the SWIM system are all trusted civil aviation information units that are admitted through the consortium blockchain mechanism. It is necessary to further improve the protocol process of PBFT consensus, improve the operational efficiency of the system, and ensure that it is suitable for the actual environment and business needs of SWIM.

6.2 Conclusion

In this paper, the security problem of cross-domain authentication of the SWIM system is solved. A cross-domain authentication model for SWIM based on the consortium blockchain is designed, including the domain, the service authentication consistency hash space, the regional authentication center consistency hash space, and the function of authenticating the domain. Meanwhile, the blockchain certificate format for cross-domain access services oriented to the SWIM consortium blockchain is designed. On this basis, a cross-domain authentication protocol for the SWIM consortium blockchain based on a consistent hash algorithm is proposed, which uses a cluster of authentication centers with virtual nodes, synchronizes the authentication mapping relationship of users between different authentication domains, designs the cutting method of the consistent hash ring for SWIM services, and realizes the dynamic load balancing of cross-domain authentication for different services. Moreover, the security analysis and performance typing show that through the deployment of the blockchain network, the management of user certificates, and the verification of cross-domain access, etc., the ability to cope with the man-in-the-middle attacks, replay attacks, Sybil attacks, and distributed denial-of-service attacks of SWIM cross-domain authentication is satisfied, and the distributed architecture can improve the security of the SWIM system; the total time of the consensus process of the user's cross-domain authentication information in the SWIM system is 1.8698 s, realizing the fast authentication of user identity across domains; this paper using the characteristics of consistent hash, the number of consistent hash virtual nodes is continuously increased, and the feasibility of cross-domain authentication load balancing is satisfied through the comparison of the average response time of authentication.

Acknowledgement: The security of SWIM is the basis of our research. We are thankful to all the researchers and applications that led us to develop a SWIM consortium blockchain cross-domain authentication system.

Funding Statement: This research was funded by the National Natural Science Foundation of China (62172418), the Joint Funds of the National Natural Science Foundation of China and the Civil Aviation Administration of China (U2133203), the Education Commission Scientific Research Project of Tianjin China (2022KJ081) and the Open Fund of Key Laboratory of Civil Aircraft Airworthiness Technology (SH2021111907).

Author Contributions: Study conception and design: L.Z, Y.H; methodology: L.Z, J.N; software: L.Z, Y.H; data collection: Y.H, K.W; analysis and interpretation of result: L.Z, Y.H; draft manuscript preparation: Y.H, L.Z. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this paper can be requested from the corresponding author upon request.

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

References

[1] Z. Liu, "In-domain user identity authentication method for system wide information management (SWIM)," in *Proc. of the 12th Int. Conf. on Machine Learning and Computing*, Shenzhen, China, pp. 479–482, 2020.

[2] L. Zhang, Z. You, K. Wang and Z. Cui, "Research on access control scheme of system wide information management based on attribute association," *Security and Communication Networks*, vol. 2022, 6181995, 2022.

[3] R. Sabatini, A. Roy, E. Blasch, K. Kramer, G. Fasano *et al.,* "Avionics systems panel research and innovation perspectives," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 12, pp. 58–72, 2020.

[4] J. Nie, X. Xu and J. Lei, "Research on potential hazards of information security in SWIM title," *Information Security Research*, vol. 7, no. 8, pp. 745–753, 2021 (In Chinese).

[5] D. Karger, E. Lehman, T. Leighton, L. Matthew, L. Daniel *et al.,* "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web," in *Proc. of the 29th ACM Symp. on Theory of Computing*, El Paso, USA, pp. 654–663, 1997.

[6] X. Wu, K. Fang and Y. Yang, "Group consistent hash data segmentation method," *Computer Engineering and Design*, vol. 37, no. 2, pp. 363–366+371, 2016.

[7] Y. Nakatani, "Structured allocation-based consistent hashing with improved balancing for cloud infrastructure," *IEEE Transactions on Parallel and Distributed Systems*, vol. 39, no. 9, pp. 2248–2261, 2010.

[8] K. Thar, S. Uilah and C. Hong, "Consistent hashing based cooperative caching and forwarding in content centric network," in *Proc. of the 16th Asia-Pacific Network Operations and Management Symp. (APNOMS)*, Taiwan, pp. 1–4, 2014.

[9] Y. Yao and X. Wang, "Web cross-domain authentication optimization scheme based on consistent hash," *Journal of Northeast Normal University (Natural Science Edition)*, vol. 45, no. 2, pp. 55–60, 2013.

[10] R. Ahmad, K. Salah, R. Jayaraman, H. Hasan, I. Yaqoob *et al.,* "The role of blockchain technology in aviation industry," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 3, pp. 4–15, 2021.

[11] W. Zou, D. Lo, P. Kochhar, X. Le, X. Xia *et al.,* "Smart contract development: Challenges and opportunities," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, 2019.

[12] X. Ma, W. Ma and X. Liu, "Cross domain authentication scheme based on blockchain technology," *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2571–2579, 2018.

[13] L. Chen, F. Xiang and Z. Sun, "Research progress of blockchain security technology based on attribute cryptosystem," *Acta Electronica Sinica*, vol. 49, no. 1, pp. 192–200, 2021.

[14] L. Xue, H. Huang, F. Xiao and W. Wang, "A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums," *IEEE Transactions on Network and Service Management*, vol. 18, no. 9, pp. 1–12, 2020.

[15] W. Wang, N. Hu and X. Liu, "BlockCAM: A blockchain-based cross-domain authentication model," in *Proc. of the IEEE Third Int. Conf. on Data Science in Cyberspace*, Guangzhou, China, pp. 896–901, 2018.

[16] Y. Zhang and B. Xing, "Cross domain authentication scheme based on multi-layer blockchain," *Computer Application Research*, vol. 38, no. 6, pp. 1637–1641, 2021.

[17] G. Ali, N. Ahmad, Y. Cao, S. Khan, H. Cruickshank *et al.,* "xDBAuth: Blockchain based cross domain authentication and authorization framework for internet of things," *IEEE Access*, vol. 8, pp. 58800–58816, 2020.

[18] G. Li, Y. Wang, B. Zhang and S. Lu, "Smart contract-based cross-domain authentication and key agreement system for heterogeneous wireless networks," *Mobile Information Systems*, vol. 2020, 2964562, 2020.

[19] M. Conti M, N. Dragoni and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.

[20] S. Pradhan, W. Sun, G. Baig and L. Qiu, "Combating replay attacks against voice assistants," in *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–26, 2019.

[21] S. Murali and A. Jamalipour, "A lightweight intrusion detection for sybil attack under mobile RPL in the Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 379–388, 2019.

[22] Q. Nasir, I. Qasse, M. Abu and A. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 9, 3976093, 2018.

[23] W. Cai, L. Yu, R. Wang, N. Liu and E. Deng, "Research on application system development method based on blockchain," *Journal of Software*, vol. 28, no. 6, pp. 1474–1487, 2017.

[24] X. Zheng, W. Feng, M. Huang and S. Feng, “Optimization of PBFT algorithm based on improved C4.5,” *Mathematical Problems in Engineering*, vol. 2021, 5542078, 2021.

[25] W. Zhang, X. Wang, W. Guo and D. He, “Efficient cross domain authentication scheme for virtual enterprises based on elliptic curve cryptosystem,” *Acta Electronica Sinica*, vol. 42, no. 6, pp. 1095–1102, 2014.

[26] G. Zhao, B. Di and H. He, “A novel decentralized cross-domain identity authentication protocol based on blockchain,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 1, pp. e4377, 2022.