Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042000

ARTICLE

EduASAC: A Blockchain-Based Education Archive Sharing and Access Control System

Ronglei Hu[1], Chuce He[1], Yaping Chi[2], Xiaoyi Duan[1], Xiaohong Fan[1], Ping Xu[1] and Wenbin Gao[1,*]

[1] Department of Electronics and Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing, 100070, China

[2] Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing, 100070, China

*Corresponding Author: Wenbin Gao. Email: gaowenbin@besti.edu.cn

Received: 14 May 2023 Accepted: 03 November 2023 Published: 26 December 2023

ABSTRACT

In the education archive sharing system, when performing homomorphic ciphertext retrieval on the storage server, there are problems such as low security of shared data, confusing parameter management, and weak access control. This paper proposes an Education Archives Sharing and Access Control (EduASAC) system to solve these problems. The system research goal is to realize the sharing of security parameters, the execution of access control, and the recording of system behaviors based on the blockchain network, ensuring the legitimacy of shared membership and the security of education archives. At the same time, the system can be combined with most homomorphic ciphertext retrieval schemes running on the storage server, making the homomorphic ciphertext retrieval mechanism controllable. This paper focuses on the blockchain access control framework and specifically designs smart contracts that conform to the business logic of the EduASAC system. The former adopts a dual-mode access control mechanism combining Discretionary Access Control (DAC) and Mandatory Access Control (MAC) and improves the tagging mode after user permission verification based on the Authentication and Authorization for Constrained Environments (ACE) authorization framework of Open Authorization (OAuth) 2.0; the latter is used in the system to vote on nodes to join requests, define access control policies, execute permission verification processes, store, and share system parameters, and standardize the behavior of member nodes. Finally, the EduASAC system realizes the encryption, storage, retrieval, sharing, and access control processes of education archives. To verify the performance of the system, simulation experiments were conducted. The results show that the EduASAC system can meet the high security needs of education archive sharing and ensure the system's high throughput, low latency, fast decision-making, and fine-grained access control ability.

KEYWORDS

Blockchain; data security; access control; smart contract

1 Introduction

With the rapid improvement of social education, many multidisciplinary and multi-form education archives have been generated involving students, educational institutions, enterprises, government departments, and other stakeholders. At the same time, data breaches occur frequently in the education field, and attack methods are increasing day by day. In January 2022, Illuminate Education, an online scoring and attendance system, was attacked, and hackers gained access to the personal database of approximately 820,000 New York City public school students. In May of the same year, a large-scale data breach occurred in Chicago Public Schools in the United States, and the data of nearly 500,000 students and 60,000 employees were leaked. Therefore, the needs of organizations and institutions for the authority, security, and comprehensiveness of education archives have increased, and the confidentiality, completeness, authenticity, and availability of educational archive-sharing systems have been improved [1]. In particular, new privacy regulations (such as the California Privacy Rights Act (CPRA) implemented in California, USA on July 01, 2023) impose stricter privacy protection obligations on organizations. The high-requirement education archive-sharing system is designed to store education archives in the ciphertext state in the storage server and control the access of users who request data. After the user access permission is approved, the storage server runs the ciphertext retrieval mechanism, thus completing the process of searching and sharing the target data in the database.

In the research work, this storage server used for data storage, ciphertext retrieval, and access control is generally designed as a third-party storage service provider. It has the advantages of large storage capacity, low cost, multi-functionality, and easy sharing. Still, as a third party of interest, it may raise problems such as low user trust, high user privacy risk, and less user control over data. The homomorphic ciphertext retrieval mechanism running on the storage server has a high algorithm, key, encrypted data security, and retrieval efficiency. Still, it has problems such as extensive computation, complex parameter sharing, and retrieval access control. The centralized access control mechanism running on storage servers can deny unauthorized access requests from illegal users and overstepped access requests from legitimate users to ensure controlled and legitimate use of system resources. Still, it has problems such as centralized power, single point of failure, high operating costs, third-party service trust, low scalability, etc. It cannot be used in application scenarios with decentralized equipment, multi-party management, and multiple power assignments [2,3].

This paper adopts a blockchain network with distributed, non-tamperable, public, and traceable features to run the access control mechanism to solve the abovementioned problems and challenges. It can provide a retrieval permission verification service for the homomorphic ciphertext retrieval mechanism on the storage server. It finally realizes a blockchain-based education archive sharing and access control system, EduASAC. The system scheme includes access control policies and an education archive-sharing framework and realizes data encryption, storage, retrieval, sharing, and access control processes in modules. The blockchain can realize voting node joining requests, algorithm parameter sharing, access policy storage, access permission verification, and record system behavior, which is a verifiable and unchangeable ledger. The storage server is responsible for storing big data education archives and running homomorphic ciphertext retrieval mechanisms. The smart contract on the chain automatically executes the behaviors, functions, and strategies that have reached consensus, responds to member nodes' join requests or access data requests, and ensures the consistency of network-wide behavior under non-manual intervention.

The significance and value of this paper are that the designed system framework can be combined with most of the proposed retrieval schemes in the homomorphic cryptographic retrieval research field and has universal applicability. We adopt blockchain technology to realize distributed access control, including the fusion of multiple access control models and the innovation of an access authorization framework. We use smart contracts to modularize and automate the system workflow, reducing errors' impact, improving operational efficiency, and reducing system costs. The system is applied to the field of education to ensure confidentiality, integrity, controllability, availability, and non-repudiation in the process of sharing education archives, which provides new solutions and ideas for various problems faced by the research work of educational systems. The main contributions of the article can be summarized as follows:

1) The EduASAC system adopts a dual-mode access control mechanism combining DAC and MAC models. It improves the access authorization framework based on the ACE authorization framework of OAuth 2.0. The system can be combined with most homomorphic ciphertext retrieval schemes to provide retrieval permission verification, retrieval parameter sharing, retrieval process records, and other services for homomorphic ciphertext retrieval mechanisms. This article details the architecture, parameters, workflow, access control, and homomorphic ciphertext retrieval modules of the EduASAC. It can encrypt, store, retrieve, share, and access education archives among educational institutions and record system parameters and user behavior in a way that prevents tampering.

2) Four smart contracts, Vote Contract, Mapping Contract, Control List Contract, and Access Control Contract are designed to automate the process of secure sharing and access control of education archives. In the scheme, joining new member nodes requires the whole network to call Vote Contract to vote, and then the vote count, result return, and record on the chain are completed through contract automation. The management of system security parameters is executed through Mapping Contract, and a variety of parameter mappings are formed and stored on the chain. Users call Control List Contract to dynamically set up Access Control List (ACL) for education archives, including adding, deleting, updating, and querying table items. The access control scheme is written into the Access Control Contract to verify permission when users request data, and the system behavior logs are recorded on the chain.

3) The theoretical analysis and experiments tested the EduASAC system. The specific work includes analyzing and comparing the system's security and performance, testing the efficiency of four smart contracts on the chain, and simulating the system's data sharing and user access control process to test the system's operational efficiency. The experimental results show that the EduASAC system has a short running time, low cost, and high application value.

The rest of this paper is organized as follows. In Section 2, we introduce the relevant work of this paper. In Section 3, we present the system architecture, system parameters, and workflow of EduASAC, and we detail the access control module and homomorphic ciphertext retrieval module. Section 4 introduces the design of four smart contracts in detail. In Section 5, we theoretically analyze the system's security and performance, test the EduASAC system's operation efficiency in the sharing and access control process of education archives, and analyze the experimental results. Section 6 concludes the paper and looks forward to future research work.

2 Related Works

This section outlines the research on blockchain application in education, focusing on innovative research on access control schemes running on different blockchain system frameworks. The advantages of blockchain technology in education are becoming increasingly prominent [4–6]. These applications include (1) ensuring the integrity and traceability of educational information on the chain [7], (2) student degree management and summative assessment of learning outcomes [8], (3) management, dissemination, and documentation of learning resources [9], (4) peer-to-peer, secure interaction and sharing of relevant people [10], (6) and decentralized control of a distributed digital ledger [11]. These advantages motivate the following summary of related research.

Literature [7] proposed an education record storage and sharing scheme EduRSS on the blockchain architecture, in which the storage server ciphertext stores the original education records and the blockchain records Hash to ensure the security and reliability of the education record storage and sharing process. The scheme design smart contract standardizes the storage and sharing behavior between nodes, and introduces cryptographic technology to encrypt and sign data. In the literature [10], a practical architecture for student certificate sharing was proposed on the underlying technology of the blockchain, using an off-chain storage mechanism and a novel privacy protection mechanism to ensure shared data security, user identity privacy, and system scalability. Finally, develop a Decentralized Application (DApp) based on Ethereum for testing. In the literature [12], a system CredenceLedger that uses blockchain to store certificate Hash was designed, which can be used to protect, share and verify student certificates. Literature [13] designed a lifelong learning log for students that can be recorded on the blockchain. That is, verifiable proof of all learning activities is recorded on the chain, which can verify the authenticity of the log and evaluate student achievement, employment options, and intelligence. In the literature [14], a blockchain-based academic degree traceability authentication system was designed, and smart contracts were used to realize the functions of data collaborative storage, user access control, and degree determination to ensure the safe and efficient operation of the system.

The above summarizes some system research work that uses the blockchain as the underlying technical architecture to serve the upper education platform. Among these works, the research on implementing access control to educational information is one of the key points which can allow or restrict users' access to education archives and improve the security of shared data. Unfortunately, the results of the survey show that there is little research work on educational systems based on blockchain networks operating access control mechanisms, and this model is mainly applied in healthcare [15], the Internet of Things [16], the Internet of Vehicles [17], and other fields.

The literature [18] was specifically designed for Internet of Things (IoT) data traffic, using blockchain technology for distributed access control, sensor data management, and auditable logging to support secure sharing and fine-grained access control of data-by-data owners. The literature [19] designs a distributed trusted access control system for IoT, which is centered on three smart contracts, namely Access Control Contract (ACC), Judgment Contract (JC), and Registration Contract (RC), and finally implemented based on the Ethernet smart contract platform. In the literature [20], IoTChain, designed based on the Object Security Architecture for the Internet of Things (OSCAR) object security model and the ACE authorization framework [21], was proposed to provide an End to End (E2E) solution for secure authorized access to IoT resources. In the literature [22], a medical application framework based on blockchain, and a cloud database solved the problem of storing and sharing big medical data in a trustless environment. The literature [23] used encryption and signature techniques to control user access to shared data pools. It used permission blockchain to ensure the security of electronic medical records, user identity legality, and traceability of behavior logs. The literature [24] proposed a cloud storage framework with fine-grained access control capability by combining Ethernet with Ciphertext-Policy Attribute-Based Encryption (CP-ABE). The data owner can set properties and access validity periods for users and invoke smart contracts to realize on-chain data storage and tracking. In the literature [25], based on the Hyperledger Fabric framework and smart contract, Attribute Based Access Control (ABAC) and decentralized, fine-grained, and dynamic access control management were implemented that can be applied to large-scale IoT environments.

The above research shows that blockchain-based access control research focuses on different system frameworks, can meet business needs in various fields, and has different research characteristics, application advantages, and realization value. The core of the research work in this paper is the framework design of a blockchain-based access control system, which is the first to serve a homomorphic ciphertext retrieval mechanism and provide retrieval permission verification service for storage servers running a homomorphic ciphertext retrieval mechanism. Therefore, the designed access control scheme is antecedence, innovation, and universality. Meanwhile, the data sharing scheme designed can solve the problems of third-party service distrust, data source uncertainty, data legitimacy, data access control, user identity privacy, as well as log integrity recording, and illegal behavior pursuit when running homomorphic ciphertext retrieval mechanisms on storage servers. In addition, the system application fully considers the user needs in the field of education, the data characteristics, and security features of education archives, so that the blockchain system has both data security sharing and access control capabilities.

3 System Model and Design

In this section, the architecture, threat model, parameters, workflow, and core functional modules of the EduASAC system are outlined in detail, which can ensure the security of educational archives and the privacy of user identities and perform efficient access control management on data.

3.1 System Architecture

The EduASAC system architecture is shown in Fig. 1, which mainly includes four major entities: government, educational institutions, storage servers, and blockchain.

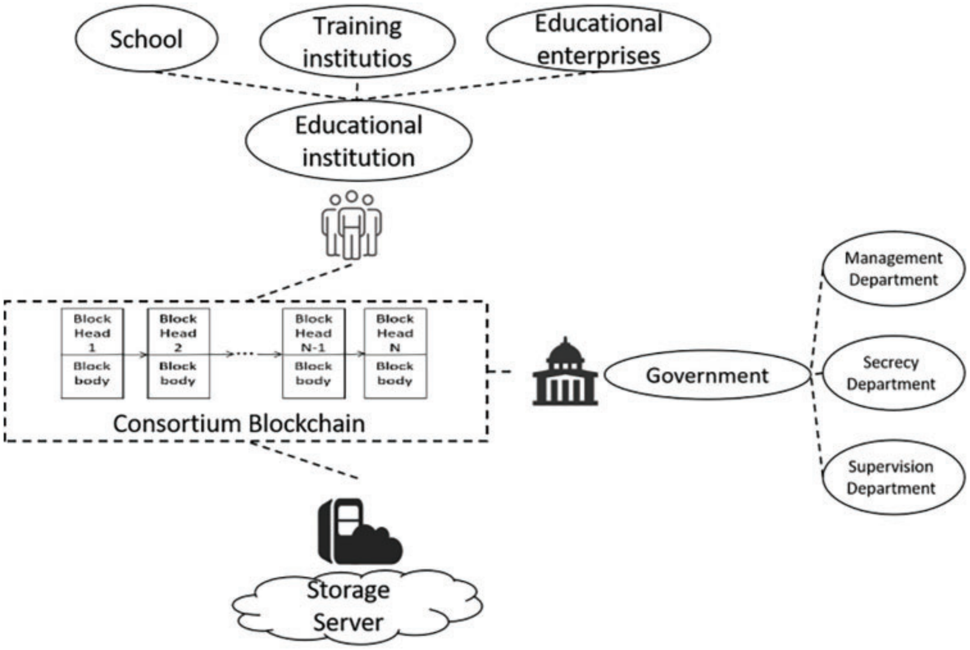

Figure 1: System architecture

1) Educational Institution: Educational institutions refer to schools, training institutions, educational enterprises, and other organizational units involved in student education. As the largest subject in the EduASAC system, each educational institution has a lifetime-bound account on the blockchain, where keys and identity information are stored. The identity information of the new educational institution is verified by the member nodes of the whole network and then joins the network after voting. Educational institutions can carry out the process of uploading, retrieving, and downloading education archives; set access rights to education archives to define the scope of data sharing; and request access tokens for target education archives.

2) Storage Server: As another main body in the EduASAC system, the storage server can provide educational institutions with services such as cryptographic storage of education archives, homomorphic cryptographic retrieval, and source data download. It can verify the identity legality of the data requestor and the authenticity of the access token on the chain and run the homomorphic ciphertext retrieval mechanism in the database after verification is passed.

3) Government: The only authoritative entity in the field of education, with the force of law and administrative capacity. The government can be divided into three major departments: management, secrecy, and supervision. Different departments have different identities, different participation links, and different roles in the system. The following describes separately:

- **Management Department:** responsible for the off-chain verification of the identity information of the nodes requesting to join; able to assign the static access permission level of the system based on the identity information; publish the verified identity information and level on the chain, and initiate Vote Contract to count the opinions of the whole network on the new nodes joining the network; record all registration behaviors and voting results on the chain.
- **Secrecy Department:** responsible for re-encryption work of ciphertext data to ensure the security of the key.
- **Supervision Department:** for illegal member nodes, illegal system behavior, and illegal data sharing. Review, determine responsibility and punish illegal targets, and maintain the security and stability of the system.

4) Blockchain: blockchain is the underlying technical architecture of the system and is responsible for promoting the network-wide voting process, verifying entity identity information, sharing system security parameters, enforcing data access control, and recording system behavior and data. The on-chain nodes are responsible for constructing new transactions from the uploaded, shared data, packaging them into blocks after verification, and recording all behaviors and data of the system as evidence of future review and accountability by the Supervision Department.

In addition to the system architecture briefly described above, to simplify the description of the scheme, this paper divides the system participants into two identity entities, i.e., Data Owners (DO) and Data Users (DU):

- DO: It has the authority to manage, control, and share education records, is the owner of interests related to the data, and can set data access permission to control the scope of sharing.
- DU: It requests targeted education archives from multiple parties according to individual needs and uses the data in various application scenarios.

3.2 Threat Model and Goals

The threat model of the EduASAC system is divided into three levels:

- **Data threat level:** The data threat level defines that the attacker performs some illegal operations on the data without the data owner's knowledge or against the data owner's will, thereby endangering the privacy of the user's identity and the value of the data.
- **Entity identity threat level:** The entity identity threat level defines that attackers join the network with illegal identities or pretend to be legal identities and bring security threats to other legitimate users.

- **System behavior threat level:** The system behavior threat level defines the system or system entity, due to its own economic interests or software and hardware failures, does not operate following pre-consensus rules and terms, which affects the stability of the system operation.

This paper aims to achieve the following threat model goals:

- The public key encryption algorithm is used to encrypt the data, and the private key is stored confidentially to ensure the security of the encrypted data; some sensitive data are stored on the chain, relying on the anti-tampering feature of the blockchain to ensure the integrity of the data.
- The authoritative entity verifies the identity information off-chain, and allows new users to join by voting to ensure the fairness and openness of entity identity information; the smart contract automatically executes the process of storing entity identity information on the chain, ensuring that there is no manual intervention so that the entire network can query that identity information.
- Record all system behaviors through smart contracts to ensure that system behaviors are auditable and traceable; the authoritative entity determines the system behavior threat level based on high-reliability records on the chain.

3.3 Access Control Model

As a core module of the system architecture, access control is based on improving the ACE authorization framework, DAC, and MAC model, with enhanced security and availability as research objectives.

Internet Engineering Task Force (IETF) ACE [21] proposes a generic framework for authentication and authorization in restricted environments-an ACE based on OAuth 2.0 [26]. DO do not have the ability to control access by themselves and need to issue tokens through third-party authorization servers to control access to sensitive, protected resources. To securely distribute Tokens with access control capabilities must establish a secure channel between users and authorization servers to encrypt data and verify the identity or use technologies such as certificates and secret sharing. At the same time, the third-party authorization server that performs access control according to the user's wishes must be honest and trustworthy to ensure the authority and authenticity of the Token. These problems make the application of the ACE framework subject to multiple environmental constraints, with low security and availability. The research work in this paper uses blockchain instead of a third-party authorization server in the ACE framework, and smart contracts are responsible for generating access Tokens and storing them securely on the chain for other users to query and verify.

DAC means that the data owner can autonomously grant or withdraw access permissions for the object requesting access to the data. The research work in this paper uses an ACL to add, change, and delete user access permissions dynamically. It combines with MAC to improve access control management's flexibility, reduce access control mechanisms' operating costs, and make the system scheme more robust and available.

MAC is a mandatory security attribute setting rules for all subjects, where the system decides whether the user can access according to the user's fixed security attributes. The research work in this paper adopts static permission levels as the fixed security attributes. They are collectively referred to as Static Access Control (SAC) below for ease of description. It can make up for the decentralized permission control ability of ACL. Because the system forcibly sets the static authority level, it has unchangeable and strong control characteristics and improves user permissions' security.

3.4 System Parameter

In this subsection, we define the designed system security parameters. The parameters include four Identity Documents (ID) of Data Owner (ID_{DO}), Data User (ID_{DU}), Data (ID_{data}), and Access Token (ID_{token}), two access control policies (P) of SAC and ACL, three mappings (M) of Verification Parameter Mapping (MV), Retrieval Parameter Mapping (MR), and Encryption Parameter Mapping (ME), four smart contracts of Vote Contract, Mapping Contract, Access Control Contract, and Control List Contract, three attributes of User, Data, and T with the access token (Token). The used symbols are described in Table 1. The above system parameters are defined as follows:

$$\mathrm{P} = \{\mathrm{SAC},\ \mathrm{ACL}\}$$

$$\mathrm{M} = \{\mathrm{MV},\ \mathrm{MR},\ \mathrm{ME}\}$$

$$\mathrm{MV} = \{\mathrm{ID}_{\mathrm{DO}},\ \mathrm{ID}_{\mathrm{data}},\ \mathrm{VP}\}$$

$$\mathrm{MR} = \{\mathrm{ID}_{\mathrm{DO}},\ \mathrm{ID}_{\mathrm{data}},\ \mathrm{C}\,(\mathrm{RK})\}$$

$$\mathrm{ME} = \{\mathrm{ID}_{\mathrm{DO}},\ \mathrm{ID}_{\mathrm{data}},\ \mathrm{C}\,(\mathrm{EP})\}$$

$$\mathrm{User} = \{\mathrm{ID}_{\mathrm{DO}}\,(\mathrm{ID}_{\mathrm{DU}})\,,\ \mathrm{PK},\ \mathrm{SK},\ \mathrm{Info},\ \mathrm{S}\}$$

$$\mathrm{Data} = \{\mathrm{ID}_{\mathrm{data}},\ \mathrm{VP},\ \mathrm{RP},\ \mathrm{EP},\ \mathrm{K},\ \mathrm{D}\}$$

$$\mathrm{T} = \{\mathrm{ID}_{\mathrm{token}},\ \mathrm{C}\,(\mathrm{RK})\,,\ \mathrm{C}\,(\mathrm{EP})\}$$

Table 1: Symbol descriptions

Name	Meaning
VP	Security parameters used in the validation process
RP	Security parameters used in the retrieval process
EP	Security parameters used in the encryption process
C (EP)	EP is encrypted by the public key of the government secrecy department
C (RK)	RP and ciphertext keywords are encrypted by the public key of the storage server
K	Keywords
C (K)	Ciphertext keywords
D	Education archive data
C (D)	Education archive ciphertext data
Info	Node identity information, which needs to be authentic and legal for all nodes to verify
S	Static access permission level
Message	Formed by Info and S Packaging

Note: Security parameters include algorithm parameters and keys required in various workflows, and their values are safe and confidential.

3.5 Workflow

As shown in Fig. 2, the entire workflow of the system is divided into four parts. This section will detail the specific steps of each part, and the symbols are shown in Table 1.

1) Register

When an educational institution joins the network, it sends a registration application to the government. As the guide node, the Government Management Department verifies the identity information of the node off-chain and publishes it on the chain, and then initiates a Vote Contract to vote for the joining request of the new node by the nodes of the whole network.

Step 1 When an educational institution applies to enter the EduASAC system, it will send Info to the government off-chain, including the institution's name, type, number, and relevant identity certificates. The Government Management Department is responsible for verifying the authenticity of the Info.

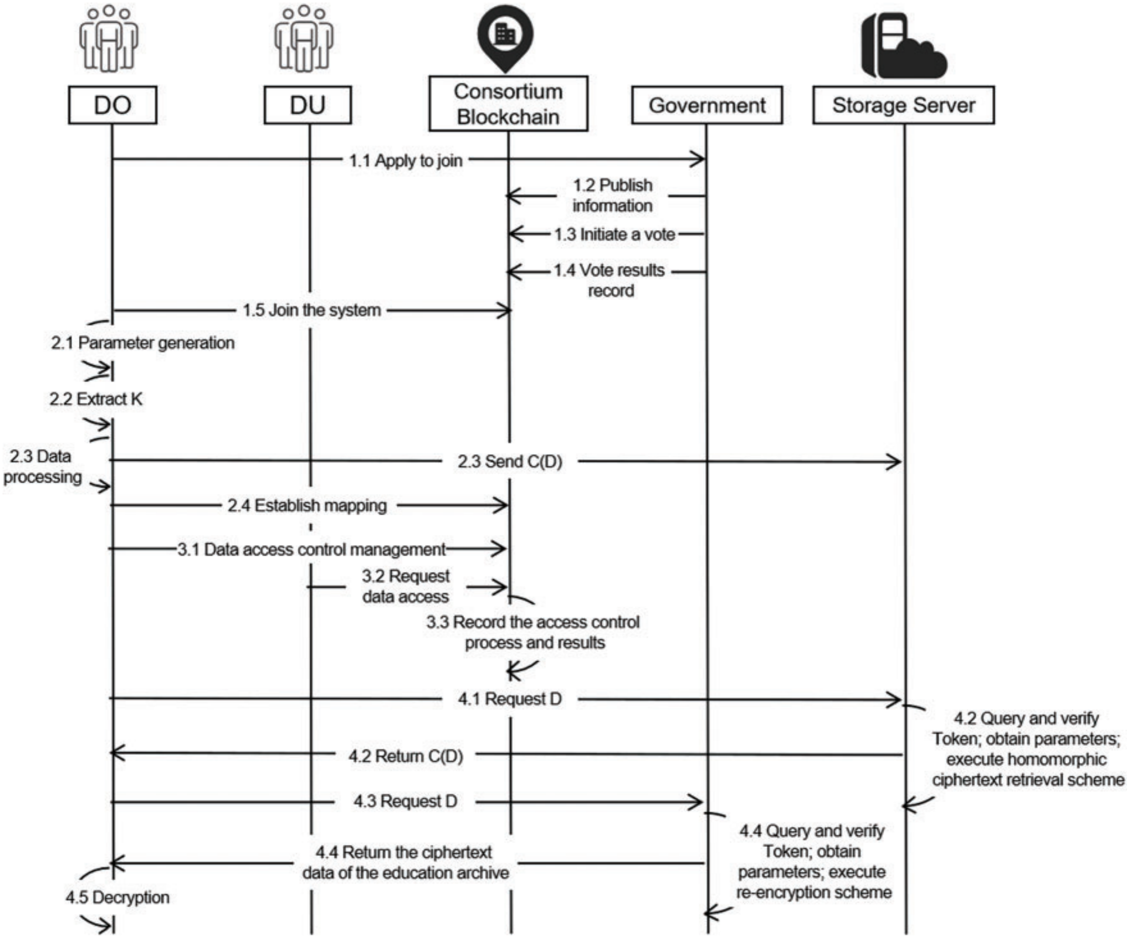

Figure 2: Workflow of EduASAC system

Step 2 After the Government Management Department approves, the static node access level S_{Info} is allocated according to Info. Pack {Info, S} into Message, forming a transaction and publishing it on the chain.

$$\mathrm{Build}\,(\mathrm{Info},\ S_{\mathrm{Info}}) \rightarrow \mathrm{Message} \tag{1}$$

Step 3 The Government Management Department calls Vote Contract to initiate an on-chain vote. The nodes of the entire network participate in the voting within the voting period, and the contract automatically counts the number of votes. If the consent vote exceeds 90% of the total votes, the contract returns the Yes result and allows the new node to join; otherwise, it returns the Refuse result and rejects the new node.

$$\text{Vote Contract (Message)} \rightarrow \text{Yes} \rightarrow \text{join} \tag{2}$$

$$\text{Vote Contract (Message)} \rightarrow \text{No} \rightarrow \text{refuse} \tag{3}$$

Step 4 Vote Contract will record the results on the chain after the voting is completed, providing valid evidence for the follow-up work such as review and accountability. Details are shown in Table 2.

Table 2: Voting result record

Node ID	Total number of votes participated	Statistical results (Yes/No)	Nodes allowed/denied to join	Voting time (start/end)
2023010	100	98/2	allowed	20230101/20230901
2023011	200	5/195	denied	20230521/20230721
...	...	...	...	...

Step 5 When a new educational institution joins the EduASAC system, it will act as a blockchain member node. Certificate Authority (CA) creates a certificate for it, generates a public key (PK) and secret key (SK), and deposits them in the wallet. PK is published to the public as a blockchain address, a unique identifier on the chain of educational institutions to participate in or link to the blockchain network. SK is secretly stored for transaction signatures or decryption of secret data.

2) Data Management

DO first processes the data with various algorithmic mechanisms for the education archives that must be uploaded to the storage server, generating parameters that participate in the sharing and access control process. Then DO invokes the Mapping Contract to establish three parameter mappings and store them securely on the chain for easy calling by other system entities when performing arithmetic operations on the data.

Step 1 DO generate parameters or keys EP, RP and VP off-chain applying them to homomorphic encryption, ciphertext retrieval, and access control processes, respectively. The VP settings are shown in Table 3.

$$\text{Build}\ (\text{ID}_{\text{DO}}) \rightarrow \{\text{VP},\ \text{RP},\ \text{EP}\} \tag{4}$$

Table 3: VP settings

Parameter	Description
0	No access control is required.
1	The access control process performs SAC only.
2	The access control process performs ACL only.
3	SAC and ACL run the access control process.
Other	Perform other access control procedures. This parameter represents other access control schemes or is used as a security parameter in a permission verification scheme.

Step 2 DO extract n (n $\geq$ 1) keywords $K_1,\ K_2,\ K_3,\cdots,K_n$ from the education archive and set static access permission level S_{data} for the education archive.

$$\text{Build}\left(\text{ID}_{\text{data}}\right) \rightarrow \text{S}_{\text{data}} \tag{5}$$

Step 3 DO homomorphically encrypts the education archive with EP and sends it to the storage server; homomorphically encrypts the keywords with the same EP; encrypts the EP with PK of the Government Secrecy Department to get C (EP); for the ciphertext keyword list and RP, encrypt them with PK of the storage server to form C (RK). This step is the data processing process, the algorithmic process runs off-chain, and the computing process can be changed according to the system scheme design.

$$\text{Encrypt}\left(\text{D}\right)_{\text{EP}} \rightarrow \text{C}\left(\text{D}\right) \tag{6}$$

$$\text{Encrypt}\left(K_1,\ K_2,\ K_3\cdots K_n\right)_{\text{EP}} \rightarrow \left\{\text{C}\left(K_1\right),\ \text{C}\left(K_2\right),\ \text{C}\left(K_3\right)\cdots\text{C}\left(K_n\right)\right\} \tag{7}$$

$$\text{Encrypt}\left(\text{EP}\right)_{\text{PK}} \rightarrow \text{C}\left(\text{EP}\right) \tag{8}$$

$$\text{Encrypt}\left(\text{C}\left(K_1\right),\ \text{C}\left(K_2\right),\ \text{C}\left(K_3\right)\cdots\text{C}\left(K_n\right),\ \text{RP}\right)_{\text{PK}} \rightarrow \text{C}\left(\text{RK}\right) \tag{9}$$

Step 4 DO invokes Mapping Contract to establish ME, MR, and MV and stores the three mapping on the chain. Different mappings will be used in different system entities and steps, effectively reducing the correlation between algorithm parameters and keys and facilitating system management and sharing.

$$\text{Build}\left(\text{ID}_{\text{DO}},\ \text{ID}_{\text{data}},\ \text{C}\left(\text{EP}\right)\right) \rightarrow \text{ME} \tag{10}$$

$$\text{Build}\left(\text{ID}_{\text{DO}},\ \text{ID}_{\text{data}},\ \text{C}\left(\text{RK}\right)\right) \rightarrow \text{MR} \tag{11}$$

$$\text{Build}\left(\text{ID}_{\text{DO}},\ \text{ID}_{\text{data}},\ \text{VP},\ \text{S}_{\text{data}}\right) \rightarrow \text{MV} \tag{12}$$

3) Access Control

The access control process of the EduASAC contains SAC, ACL, or other access control mechanisms to realize DO fine-grained access control education archives.

Step 1 DO calls Control List Contract as needed to set data access permissions on the chain. The process will be matched to the VP values, as shown in Table 2.

$$\text{Build}\left(\text{ID}_{\text{DO}},\ \text{ID}_{\text{data}}\right) \rightarrow \text{P} = \left\{\text{SAC},\ \text{ACL}\right\} \tag{13}$$

- SAC: the S_{data} of the data has been set when DO creates the MV of the education archive. The S_{Info} of educational institutions has been configured by government management according to the Info of educational institutions and voted by the member nodes of the whole network.
- ACL: realize adding, updating, deleting, and querying dynamic access list. The dynamic access list is divided into Dynamic Allowed Access List (DAAL), Dynamic Denied Access List (DDAL):
    a. DAAL: can be used in the separate ACL process, allowing only the allowed objects set by DO in the list to share data; can be used in the ACL and SAC co-working process, which is for users with low static access permission level, and DO adds their information to DAAL to give them additional access permissions. See Table 4.
    b. DDAL: serves SAC and will allow DO to deny access to users with high static access permission levels. See Table 5.

Table 4: Dynamic allowed access list

$\mathrm{ID}_{\mathrm{data}}$	$\mathrm{ID}_{\mathrm{DO}}$	PK of the user allowed accessing	Validity period
20239111	20230001	AAAC3Nzax1459dAeRfs . . .	20231231
20239112	20230002	GBn1eGHL5478E9F457R . . .	20240101
. . .	. . .	. . .	. . .

Table 5: Dynamic denied access list

$\mathrm{ID}_{\mathrm{data}}$	$\mathrm{ID}_{\mathrm{DO}}$	PK of the user denied accessing	Validity period
20230312	20231001	RdnfG45efe85Gfge4584G . . .	20240501
20230313	20231002	L14Gh54eGi8feges87g4fg . . .	20241231
. . .	. . .	. . .	. . .

Step 2 DU calls the Access Control Contract to request access to the target data. The contract validates DU access permissions according to DO's access control management of the data. When the access permissions are verified, the contract generates Tokens and forms blockchain transactions stored on the chain.

$$\text{Access Control Contract}\left(\mathrm{ID}_{\mathrm{DU}},\ \mathrm{ID}_{\mathrm{data}}\right) \rightarrow \text{Token} \tag{14}$$

Step 3 Access Control Contract will record the access control process and results for review and accountability by the Government Supervision Department. The access control information records are shown in Table 6.

Table 6: Access control information record

$\mathrm{ID}_{\mathrm{DU}}$	$\mathrm{ID}_{\mathrm{data}}$	Access control policy	Whether the Token has been issued ($\mathrm{ID}_{\mathrm{token}}$)	Request/Response time (Request, issue Token time)
20232001	20235311	SAC	2023645001	20230901/20230901
20232002	20236312	ACL	2023645002	20230912/20230912
. . .	. . .	. . .	. . .	. . .

4) Data Sharing

Step 1 DU sends $\mathrm{ID}_{\mathrm{DU}}$ and $\mathrm{ID}_{\mathrm{data}}$ to the storage server on the chain, requesting C (D).

Step 2 The storage server queries the Token and verifies the validity according to the correspondence of $\mathrm{ID}_{\mathrm{DU}}$, $\mathrm{ID}_{\mathrm{data}}$, and $\mathrm{ID}_{\mathrm{token}}$ stored on the chain. After passing, SK is used to decrypt some parameters of the Token to obtain RP, C (K); the homomorphic ciphertext retrieval mechanism is run to get C (D) returned to DU.

$$\text{Decrypt}\left(\mathrm{C}\left(\mathrm{RK}\right)\right)_{\mathrm{SK}} \rightarrow \left\{\mathrm{C}\left(K_1\right),\ \mathrm{C}\left(K_2\right),\ \mathrm{C}\left(K_3\right)\cdots\mathrm{C}\left(K_n\right),\ \mathrm{RP}\right\} \tag{15}$$

$$\text{Retrieval}\left(\mathrm{C}\left(K_1\right),\ \mathrm{C}\left(K_2\right),\ \mathrm{C}\left(K_3\right)\cdots\mathrm{C}\left(K_n\right)\right)_{\mathrm{RP}} \rightarrow \mathrm{C}\left(\mathrm{D}\right) \rightarrow \mathrm{DU} \tag{16}$$

Step 3 DU sends C (D) through the off-chain secure channel to the Government Secrecy Department to request a re-encryption process to obtain D. This step aims to reduce the range of key sharing and improve key security.

Step 4 According to the correspondence of $\mathrm{ID}_{\mathrm{DU}}$, $\mathrm{ID}_{\mathrm{data}}$ and $\mathrm{ID}_{\mathrm{token}}$ stored on the chain, the Government Secrecy Department queries the Token and verifies the validity; after passing, decrypts some parameters of the Token to obtain EP using SK; runs the re-encryption process and returns the DU decryptable education archive. The re-encryption process is that the Government Secrecy Department Decrypts C (D) with EP, then queries PK of DU on the chain, and re-encrypts the education archive back to DU.

$$\mathrm{Decrypt}\,(\mathrm{C}\,(\mathrm{EP}))_{\mathrm{SK}} \rightarrow \mathrm{EP} \tag{17}$$

$$\mathrm{Decrypt}\,(\mathrm{C}\,(\mathrm{D}))_{\mathrm{EP}} \rightarrow \mathrm{D} \tag{18}$$

Step 5 DU uses SK to decrypt the education archives of the cryptographic state.

3.6 Homomorphic Ciphertext Retrieval Module

The ciphertext retrieval scheme [27] based on a homomorphic encryption algorithm was first proposed by BONEH [28], and it is based on Difficulties in cryptography. The strategy is to obtain retrieval results by comparing homomorphic ciphertext and ciphertext keywords with high data security and retrieval efficiency. The EduASAC system proposed in this paper can be used with most homomorphic ciphertext retrieval schemes running on storage servers to achieve decentralized fine-grained access control and efficient data sharing.

Homomorphic encryption schemes generally consist of four Probabilistic Polynomial Time (PPT) algorithms [29]. The specific content is as follows:

1) Key generation algorithm (KeyGen). Input security parameters and public parameters that meet other actual needs and can output the encryption key (public key), the decryption key (private key), and the public key used for homomorphic ciphertext calculation.
2) Encryption algorithm (Encrypt). Input the plaintext and encryption key and output the ciphertext result of the encryption operation. Among them is a one-to-one correspondence between plaintext and ciphertext, and the ciphertext results obtained by encrypting the same plaintext are different.
3) Decryption algorithm (Decrypt). Input the ciphertext and decryption key and output the plaintext result of the decryption operation.
4) Homomorphic computing algorithm (Evaluate). It is a homomorphic correctness verification algorithm, which does not involve any system modules in this paper, and the specific process is omitted.

In the workflow shown in Fig. 2, Mark 2.1 corresponds to KeyGen (). DO generates keys and security parameters of the homomorphic encryption algorithm. Mark 2.2 is DO extracts keywords from archives. Mark 2.3 corresponds to Encrypt (), DO encrypts education archives and keywords and uploads them to the storage server and blockchain. Mark 4.4 corresponds to Decrypt (), the government department decrypts education archives, performs re-encryption operations, and then shares them with DU. Mark 4.2 corresponds to the homomorphic ciphertext retrieval algorithm Retrieval (), which is run independently by the storage server. The above steps are all run off-chain. It can effectively reduce the amount of computation on the chain, improve the system's scalability, enable the system framework to be used in conjunction with most homomorphic ciphertext retrieval schemes, and meet the different needs of different retrieval schemes for access control.

The EduASAC system provides the following services for the homomorphic ciphertext retrieval mechanism:

1) For the key and security parameter EP of the homomorphic encryption algorithm, it is packaged to form ME and stored on the chain.
2) For the ciphertext keywords after homomorphic encryption, they are packaged together with the retrieval security parameter RP to form MR and stored on the chain.
3) For the educational archives stored on the storage server, the access control management of SAC and ACL is carried out based on the blockchain network, and the retrieval permission can be verified before running the homomorphic ciphertext retrieval mechanism.
4) Blockchain implements access control on the DU that requests the retrieval data service, calls the contract to verify the DU's retrieval permission, and automatically generates a Token and stores it on the chain. Token indicates the access permission of DU and provides corresponding parameters for the subsequent retrieval and decryption process.

The above content can be set according to user needs and the homomorphic ciphertext retrieval scheme. For example, the generation algorithm of the key and security parameter EP is set according to the homomorphic encryption scheme; when the homomorphic ciphertext retrieval process does not require additional security parameters, there is no RP setting; DO implement different levels of access control for different education archives and users.

4 Smart Contract Design

The EduASAC system uses smart contract technology to control the system's workflow, that is, the process of education archive sharing and access control. This section details four kinds of smart contracts: Vote Contract, Mapping Contract, Control List Contract, and Access Control Contract, and their associated algorithms and logic interfaces.

4.1 Vote Contract

The Government Management Department invokes the Contract and can initiate the voting of the entire network nodes on the educational institutions applying to join the network. After the voting, the Contract will automatically count the votes, return the information of allowing or rejecting the joining request, and record the voting results in a table on the chain, see Table 2 in the previous section. The specific content of the Contract is as follows:

VoteInstitution(): As shown in Algorithm 1, member nodes vote before the voting deadline. Info is the identity information of the voting node. Nodes vote for the Info included in the AccessmemberMap mapping and against the Info in the RefusememberMap mapping. Both mapping records will be uploaded to the chain. The two-mapping records deadline is the voting period, a fixed value stipulated and released by the Government Management Department on the chain.

Algorithm 1: VoteContract. VoteInstitution(): Initiate member node voting process

Input: *Info* is the voting node information, *deadline* is the voting deadline
Output:

```
If $time(now) > deadline$ then
    return false;
```

(Continued)

Algorithm 1 (continued)

```
end if
If $Info$ already exists then
  return false;
end if
If access the new institution join the Blockchain then
  $AccessmemberMap \leftarrow Info$
  return true;
else
  $RefusememberMap \leftarrow Info$
  return true;
end if
```

Algorithm 2: VoteContract. VoteCounting(): Count the votes

```
Output: $boolean$
$Yes = 0$, $No = 0$, $All = 0$
For $member$ in $AccessmemberMap$ do
  $Yes + = 1$
End for
For $member$ in $RefusememberMap$ do
  $No + = 1$
End for
$All = Yes + No$
If $Yes/All > 9/10$ then
  return true;
else
  return false;
end if
```

Algorithm 3: VoteContract. institutionMap(): Establish the mapping of educational institution information

```
Input: $Info$ is the new institution information, $ID$
Output:
If the sender's $BlockchainAddress$ can not correspond to $ID_{DO}$ then
  return false;
end if
  $institutionMap\,[ID]$ .add $(Info)$
return true;
```

```
typedefine <Struct> Info{
  ID: String.
  Publickey: String.
  Verification: String.
  Level $(S_{Info})$: String.}
```

VoteCounting(): As shown in Algorithm 2, after the voting deadline, the votes are counted. When the approval votes account for more than 90% of the total votes, new nodes are allowed to join. Otherwise, the joining request is rejected. The percentage of votes allowed to join can be changed according to actual needs.

institutionMap(): As shown in Algorithm 3, this function describes the process of establishing a mapping between ID and Info, which is executed by newly joined member nodes to initialize their official information and provide parameter records for the subsequent process of verifying user identity.

isExitInstitution(): This function checks whether the educational institution applying to join already exists in the network to prevent the internal nodes from sending many requests to affect the system's proper operation.

addVoteCase()/*queryVoteCase*(): Upload or query events that require member nodes to vote, including the id, name, content, time limit, and the number of votes in favor and against. In the initial state, the number of votes in favor and against is 0; after the voting, this value is the final result provided to *VoteCounting*() for calculation.

4.2 Mapping Contract

This Contract is invoked by DO, who uploads education archives and hopes to share them with other users, to establish three parameter mapping: ME, MR, and MV. It can associate DO identity information, education archives information, and parameters required for system work and serve the homomorphic ciphertext retrieval, access control, and re-encryption process. DO initiates the Contract, and its blockchain address needs to match the input $\mathrm{ID}_{\mathrm{DO}}$ so that the Contract can normally execute the process of establishing, updating, deleting, and querying parameter mapping. The specific content of the Contract is as follows.

Algorithm 4: MapContract. initVerifyMap(): Initialize validation parameter mapping

```
Input: $ID_{DO}$, $ID_{data}$, $VP$ and $S_{data}$
Output: boolean
If $ID_{DO}$ and $ID_{data}$ is NULL then
  return false;
end if
If the sender's $BlockchainAddress$ can not correspond to $ID_{DO}$ then
  return false;
end if
For ($skey$ in keys($institutionMap$)) do
  If $skey == ID_{DO}$ then
    For ($dataID$ in keys($VerifyMap$)) do
      If $dataID == ID_{data}$ then
        return false;
      end if
    End for
    $Verifypara \leftarrow$ initVerifypara ($ID_{DO}, VP$)
    $VerifyMap\,[ID_{data}]$ .add ($Verifypara$)
```

(Continued)

Algorithm 4 (continued)

```
            **return** true;
        **end if**
    **End for**
    **return** false;
```

Algorithm 5: MapContract. updateEncryptMap(): Updates the encryption parameter mapping

```
**Input:** $ID_{DO}$, $ID_{data}$ and $C(EP)$
**Output:** *boolean*
    **If** $ID_{DO}$ and $ID_{data}$ is NULL **then**
        **return** false;
    **end if**
    **If** the sender's *BlockchainAddress* can not correspond to $ID_{DO}$ **then**
        **return** false;
    **end if**
    **For** (*skey* **in** keys(*institutionMap*)) **do**
        **If** $skey == ID_{DO}$ **then**
            **For** (*dataID* **in** keys(*EncryptMap*)) **do**
                **If** $dataID == ID_{data}$ **then**
                    $EncryptMap[ID_{data}]$.update $(C(EP))$
                    **return** true;
                **end if**
            **End for**
        **end if**
    **End for**
    **return** false;
```

Algorithm 6: MapContract. deleteRetrievalMap(): Deletes the retrieval parameter mapping

```
**Input:** $ID_{DO}$, $ID_{data}$
**Output:** *boolean*
    **If** $ID_{DO}$ and $ID_{data}$ is NULL **then**
        **return** false;
    **end if**
    **If** the sender's *BlockchainAddress* can not correspond to $ID_{DO}$ **then**
        **return** false;
    **end if**
    **For** (*skey* **in** keys(*institutionMap*)) **do**
        **If** $skey == ID_{DO}$ **then**
            **For** (*dataID* **in** keys(*RetrievalMap*)) **do**
                **If** $dataID == ID_{data}$ **then**
                    $RetrievalMap[ID_{data}]$.remove
                    **return** true;
```

(Continued)

Algorithm 6 (continued)

```
            end if
        End for
    end if
End for
return false;
```

Algorithm 7: MapContract. queryRetrievalMap(): Queries the retrieval parameter mapping

```
Input: $ID_{DO}$, $ID_{data}$
Output: Parameter mapping or boolean
If $ID_{DO}$ and $ID_{data}$ is NULL then
    return false;
end if
If the sender's BlockchainAddress can not correspond to $ID_{DO}$ then
    return false;
end if
For (skey in keys(institutionMap)) do
    If $skey == ID_{DO}$ then
        For (dataID in keys(RetrievalMap)) do
            If $dataID == ID_{data}$ then
                $Map == RetrievalMap\,[ID_{data}]$ .query
                return Map;
            end if
        End for
    end if
End for
return false;
```

initVerifyMap()/*initRetrievalMap*()/*initEncryptMap*(): Establish validation parameter mapping, retrieval parameter mapping and encryption parameter mapping process, as shown in Algorithm 4. The same input parameters of the three functions are the IDs of DO and data, and the different input parameters are VP and S_{data}, C (RK), and C (EP).

typedefine <**Struct> VerifyPara**{

ID_{DO}: String.

ID_{data}: String.

Validation parameter: String.

Level (S_{data}): String.}

typedefine <**Struct> EncryptPara**{

ID_{DO}: String.

ID_{data}: String.

Encryption parameter: String.}

typedefine <**Struct> RetrievalPara**{

ID_{DO}: String.

ID_{data}: String.

Retrieval parameter: String.}

updateVerifyMap()/*updateRetrievalMap*()/*updateEncryptMap*(): It is the same as the input of establishing the mapping function, and the old mapping is overwritten with ID_{data}, as shown in Algorithm 5.

deleteVerifyMap()/*deleteRetrievalMap*()/*deleteEncryptMap*(): Need to input the IDs of DO and data to delete the corresponding parameter mapping, as shown in Algorithm 6.

queryVerifyMap()/*queryRetrievalMap*()/*queryEncryptMap*(): Query validation parameter mapping, retrieval parameter mapping and encryption parameter mapping procedures, as shown in Algorithm 7. Need to input the IDs of DO and data to query the parameter mapping and return the corresponding parameter or error reminder information.

4.3 Control List Contract

The role of this Contract is for the DO to perform access control management on shared education archives and call the Contract to manage and maintain two types of lists on the chain, namely the DAAL and the DDAL. As the contract initiator, DO's blockchain address needs to match the input ID_{DO}, and then the Contract can be called to add, delete, and modify the list.

addAllowList()/*updateAllowList*()/*deleteAllowList* ()/*queryAllowList* (): This function is to add, delete, modify, and query table items in the DAAL.

addDenyList()/*updateDenyList*()/*deleteDenyList*()/*queryDenyList*(): This function is to add, delete, modify, and query table items in the DDAL.

4.4 Access Control Contract

This Contract is the core of the system access control process and is used to verify the access permissions of the person who initiated the Contract. The Contract is designed according to the EduASAC system scheme, and the verification parameter mapping corresponding to the target data is queried on the chain. According to VP settings, execute the related access control process to verify DU access permissions. After the verification is passed, a Token is generated for the legal DU and stored on the chain, which can be used for subsequent operation of the homomorphic ciphertext retrieval mechanism and re-encryption process.

checkAccess(): Check DU access permissions, as shown in Algorithm 8. The input is the IDs of DU, data, and DO. A Token is generated if the verification is successful; otherwise, empty data returns to indicate that the verification failed. The access control mechanism adopted by the system will be based on the VP value set by DO. See Table 3 in the previous section.

- If VP is 0, no verification permission is required to access data, and the contract directly generates a Token.
- If VP is 1, only the SAC mechanism is required to access the data. The contract compares the static permission level S of the DU and the target data and allows access when the DU level is high. The comparison algorithm can change according to the design of the actual scheme, and choosing an algorithm with higher security and efficiency is recommended.
- If VP is 2, only the ACL mechanism is needed to access the data. The contract queries the DAAL item of the target data and allows access when there is DU information.

- If VP is 3, it indicates that SAC and ACL mechanisms are required to access data. The contract compares the static permission level S of the DU and the target data: When the DU level is high, query the DDAL item of the target data and deny access if there is DU information; when the DU level is low, query the DAAL item of the target data, and allow access if there is DU information.
- If VP is other values, the scheme will implement other access control procedures. VP represents other access control schemes or is used as a parameter in the access permission verification algorithm. The smart contract still performs the verification process, and the corresponding algorithm code needs to be written into the contract.

deleteToken()/*queryToken*(): Delete or query the Token already stored on the chain.

Algorithm 8: Access Control Contract. checkAccess(): Check the user's access rights

```
Input: $ID_{DO}$, $ID_{DU}$ and $ID_{data}$
Output: $Token$ or $null$
$K = 0, D = 0$
For ($skey$ in keys($institutionMap$)) do
  If $skey == ID_{DO}$ then
    $K = 1$
  end if
End for
For ($dataID$ in keys($VerifyMap$)) do
  If $dataID == ID_{data}$ then
    $D = 1$
  end if
End for
If $K = 0$ or $D = 0$ then
  return $null$;
end if
$ver = VerifyMap[ID_{data}].Validation\ parameter$
$L_1 = institutionMap[ID_{DU}].Level$
$L_2 = VerifyMap[ID_{data}].Level$
If $ver == 0$ then
  goto generateToken
end if
If $ver == 1$ then
  If $L_1 > L_2$ then
    goto generateToken
  else
    return $null$;
  end if
end if
If $ver == 2$ then
  If $ID_{DU}$ in DAAL then
    goto generateToken
  else
    return $null$;
```

(Continued)

Algorithm 8 (continued)

```
  end if
end if
If $ver == 3$ then
  If $L_1 < L_2$ then
    If $ID_{DU}$ in DAAL then
      goto generateToken
    else
      return $null$;
    end if
  else
    If $ID_{DU}$ in DDAL then
      return $null$;
    else
      goto generateToken
    end if
  end if
end if
If $ver$ displays other parameters then
  Implement other research-designed access control schemes
end if
generateToken:
For ($dataID$ in keys($RetrievalMap$)) do
  If $dataID == ID_{data}$ then
    $Ret = RetrievalMap\,[ID_{data}]\,.Retrieval\ parameter$
    For ($dataID$ in keys($EncryptMap$)) do
      If $dataID == ID_{data}$ then
        $Cipher = EncryptMap\,[ID_{data}]\,.Encrypt\ parameter$
        generateID:
        $ID_{Token}$ = RadomID ($ID_{data}$)
        For ($tokenID$ in keys($TokenMap$)) do
          If $tokenID == ID_{Token}$ then
          goto generateID
        end if
        End for
        $Token :< ID_{Token},\ Ret,\ Cipher >$
        $TokenMap\,[ID_{Token}]\,.\ add\ (ID_{DU})$
        return $Token$;
      end if
    End for
  end if
End for
return $null$;
```

5 Implementation and Evaluation of EduASAC

5.1 Security Analysis

According to the threat model mentioned in Section 3.2, this section theoretically analyzes the specific solutions proposed by the EduASAC system for data threat, entity identity threat, and system behavior threat.

1) Data security on the chain

The EduASAC system uses the blockchain network to store and share the data used or generated during the system's work. It deploys the smart contract that regulates the system's behavior on the chain to record the data operation process and ensure the data's integrity, correctness, and validity. Table 7 below is the security data stored on the chain.

Table 7: Secure data on the chain

Notation	Description
Token	Each system entity can determine whether the data user has access permissions according to the valid Token and then call the security parameters in the Token according to the execution mechanism.
P	Running SAC and ACL mechanisms on the blockchain can prevent attackers from illegally tampering with the access control mechanism, resulting in abnormal operation of the system.
ME/MR/MV	The mapping is stored on the chain to ensure integrity, which is convenient for users to query and share.
VP/C(RK)/ C(EP)	The contract queries the chain and extracts it as one of the Token's parameters during token generation. The homomorphic ciphertext retrieval scheme implemented on the storage server uses C (RK), which can prevent unauthorized system entities from accessing educational archives illegally through other means after having the correct ciphertext keywords; C (EP), which is used in the re-encryption process performed on Government Secrecy Departments to improve key security. VP denotes the user's access control management of data.
User	Part of the information in user attributes is stored on the chain so that system entities can verify user identity information on the chain.
Data	Part of the information in the data attributes is stored on the chain so that system entities can query other associated data through the data information.
T	Part of the information in the Token attribute is stored on the chain so that system entities can query the corresponding Token information to ensure its validity and correctness.
S	The static access level S of shared data and system entities are publicly stored on-chain. It is set by government management and data owners, respectively, and the source of parameters is authoritative. Based on the immutable feature of blockchain, it prevents participants from illegally changing permission levels to meet their interests. It facilitates system entities to check users' static access permission levels on the chain and improves operational efficiency.

One data item is not listed in the above list, i.e., all system workflow records stored on the chain. These records will serve as evidence of dishonest behavior by system entities and reduce the impact of malicious tampering of execution results or changes to the execution process by system entities. It can also provide extremely reliable data records for authorities' subsequent review and accountability process.

2) **Parameter security**

The generation, storage, and sharing of security parameters in the EduASAC system are all done by DO alone without third-party participation. Each workflow of the system is modularized, and different system working mechanisms require different security parameters, which can meet the security requirements of different mechanisms. The security parameters required by a system entity to run a certain mechanism have been encrypted by the asymmetric key of the system entity and stored on the chain in a ciphertext state.

a) The access control mechanism needs VP when verifying access permissions. Because it does not involve the security of the data itself, the access control policy setting of the data by DO, therefore, VP does not need to be stored in a ciphertext state and can be viewed by any system entity.
b) The homomorphic ciphertext retrieval mechanism requires RP and ciphertext keywords to be used when retrieving data. When a data owner grants permission to a storage server to share data, the SK of that storage server is used to encrypt the retrieved security parameters. It improves the control of DO over the data, ensures that the data is not shared by unauthorized third-party storage servers, and improves the security of the homomorphic ciphertext retrieval mechanism.
c) The re-encryption mechanism requires EP to be used when decrypting the data, which are critical keys to protect the data security and cannot be obtained by third-party service providers. Therefore, DO encrypts it with SK of the authoritative Government Secrecy Department and then uploads it to the chain in a ciphertext state. It dramatically reduces the scope of sharing critical keys and improves the security of education archives.

3) **Security of system entity identity**

a) At the time of registration, the system entity's identity information must be verified off-chain by the Government Management Department. After verification, the identity information will be published on the chain and voted on by the whole network. The opinions will be counted before deciding whether to allow them to join the EduASAC system.
b) The identity information, public key, voting process, and results of newly joined nodes are recorded on the chain for network-wide nodes to verify node identity, obtain SK and review the voting process. The private key is kept independently by the user and is not known by any system entity to ensure the secure operation of the system workflow.
c) The blockchain address of the contract initiator needs to match the input ID to ensure the authenticity and legitimacy of the participant's identity in the process of parameter mapping setting, data access permission setting, and request permission verification.
d) When DU requests retrieval and re-encryption of education archives from storage servers and the Government Secrecy Department, respectively, they need to verify the entity's identity information in the chain based on its ID.

5.2 Performance Comparison and Testing

This section explains the research characteristics, application advantages, and realization value of the EduASAC system by comparing the performance of related research schemes and conducting simulation tests.

1) Test environment

The performance test environment is shown in Table 8.

Table 8: Simulation environment

CPU	i7-10750H 2.60 GHz
Operating system	CentOS7
Blockchain platform	Hyperledger Fabric 1.4.3
Blockchain organizations and nodes	OrdererOrgs(1 Orderer node) PeerOrgs(Org1 and Org2 two organizations, 4 peer nodes)
Docker	18.06.3
Docker-compose	1.23.2
Smart contract development platform/language	Visual Studio Code/go 1.8.1
Drawing tool	Excel/Visio
Test framework	Caliper Benchmarks 0.2.0

2) Performance analysis

Table 9 below will use the designed EduASAC system as the base point to compare the data characteristics of the shared system under different storage modes.

Table 9: Comparison of data characteristics

Data characteristics	Local database (paper carrier)	Centralized database	Distributed database	Cloud	Blockchain	Our scheme
Confidentiality	×	×	×	√	√	√
Integrity	×	×	×	×	√	√
Authenticity	√	√	×	×	×	√
Share whether a third party is involved	×	√	×	√	×	√
Sharing scope	Small	Small	Big	Big	Big	Big
Share efficiency	Low	Low	High	High	High	High

(Continued)

Table 9 (continued)

Data characteristics	Local database (paper carrier)	Centralized database	Distributed database	Cloud	Blockchain	Our scheme
Data status	Handwritten paper version	Plaintext storage	Plaintext storage	Ciphertext storage	Ciphertext storage	Ciphertext storage
Manual identification required	√	√	√	×	×	×
Access control	×	√	√	√	×	√
Revocable or not	×	×	√	√	×	×
Easy to destroy	Easy	Easy	Hard	Hard	Hard	Hard

The system solution in this paper uses a combination of storage server and blockchain to store different data based on two different storage models, centralized and decentralized ledger, to jointly complete the work of sharing the data. As shown in the above table, the blockchain is a distributed structure, an open platform, without third-party participation and user access control. At the same time, because the authenticity of pre-chain data cannot be verified, there is a security threat of legitimate users uploading illegal data. There are third-party government departments with the force of law and administrative capacity in the EduASAC system, which can guarantee data security, verify data authenticity, and protect user identity privacy. Therefore, the advantages of system data characteristics are significant, and the social application value is greater. Table 10 below compares the research schemes of other blockchain education systems, highlighting the uniqueness of the scheme in this paper.

Table 10: Comparison of system characteristics

System characteristics	[7]	[10]	[12]	[13]	[14]	Our scheme
Voting nodes join	**Y**	**N**	**N**	**N**	**N**	**Y**
On-chain/off-chain storage	**Y**	**Y**	**N**	**N**	**Y**	**Y**
Multiple access control mode	**N**	**N**	**N**	**N**	**N**	**Y**
On-chain access grant	**Y**	**N**	**Y**	**Y**	**Y**	**Y**
Smart contract	**Y**	**Y**	**N**	**Y**	**Y**	**Y**
Homomorphic ciphertext retrieval application	**N**	**N**	**N**	**N**	**N**	**Y**

By comparing the system's characteristics, it can be concluded that there is originality in the research of EduASAC system. It can be combined with most homomorphic ciphertext retrieval schemes. It adopts on-chain authorization and off-chain storage, integrates multiple access control modes, and realizes sharing and access control of educational archives based on smart contracts. Table 11 compares other system research schemes that run access control policies on the blockchain.

Table 11: Solution performance comparison

Scheme	Technology/Application background	Access control policy	Security	Efficiency	Function
[18]	Based on a publicly verifiable blockchain, a centralized trusted institution is used for access control management. It is a system design scheme for IoT data traffic.	Access to the data stream is granted using the stream identifier and the public key address of the service.	Security based on data signature; data security is guaranteed through consensus protocols in the blockchain and its decentralization features.	During access control, the request throughput of the storage system is reduced by 10%; distributed storage systems with more than 1000 nodes experience a two times deceleration.	The location-aware decentralized storage system managed by blockchain technology can promote the storage of time-series IoT data storage at the network edge, enabling secure and flexible access control management.
[19]	Based on blockchain smart contract technology. It enables distributed and trusted access control for IoT systems.	Static access permissions verification based on predefined policies. Dynamic access permissions verification by checking the behavior of the subject.	The system records user misbehavior on certain resources and gives appropriate penalties.	No efficiency analysis.	The system provides functions to add, update and delete access control policies; provides methods to judge misbehavior and returns corresponding penalties after receiving misbehavior reports; can register IoT device information and can perform adding, updating, or delete operations.

(Continued)

Table 11 (continued)

Scheme	Technology/Application background	Access control policy	Security	Efficiency	Function
[22]	A medical application framework based on blockchain and cloud database; based on blockchain smart contract technology. It solves the problem of storing and sharing big medical data in a trustless environment.	A public key signature is used to verify the legitimacy of access requests.	Smart contracts and access control mechanisms can effectively track the behavior of data and realize the traceability and audit of data sources; they can revoke the access permissions of violators and minimize the risk of data privacy.	Test the time required for the transactions to analyze the latency of the system when different numbers of requesters send requests or different numbers of cloud service providers receive requests. The results show that the system is more efficient.	The system provides data source auditing and access control for big medical data that needs to be shared in the cloud database. All operations performed on the system are recorded on the chain in a tamper-proof manner, and access to the offending entity can be revoked.
[23]	The operation mechanism is based on the permission blockchain. The system leverages the blockchain's invariance and built-in autonomy features to address access control-related issues for sensitive data in cloud databases.	Secure encryption technology (encryption and digital signatures) ensures access control to sensitive shared data pools.	Cryptographic key sets ensure that security-related tasks in the system are performed correctly. Keys are applied to the user authentication process.	Test the number of user transactions on different chains for different lengths of time and the final amount of blocks generated. The system is a lightweight blockchain that is more efficient compared to Bitcoin.	The system allows users to access electronic medical records in a shared database after their identity and encryption key have been authenticated.

(Continued)

Table 11 (continued)

Scheme	Technology/Application background	Access control policy	Security	Efficiency	Function
[24]	Based on the Ethernet blockchain, cloud-based servers store data. It solves the security problem of key management controlled by a third party.	Ciphertext Policy Attribute-Based Encryption (CP-ABE).	Attribute sets have access cycles. Key distribution is decentralized and uses the Diffie-Hellman key exchange protocol to negotiate encryption keys. Smart contract invocations are restricted to specific users. Blockchain logs have privacy-preserving features.	The access control algorithm running on the contract has the same efficiency as the original one before the improvement. The system is based on the blockchain and is more efficient than the original solution.	The system implements a decentralized cloud storage framework with fine-grained access control for the stored data.
[25]	Based on Hyperledger Fabric framework; based on blockchain smart contract technology. It can be applied to large-scale IoT environments.	Attribute-based access control, and decentralized, fine-grained, and dynamic access control management.	Security based on blockchain Proof of Work (PoW) consensus.	By simulating concurrent access to the system by multi-threaded clients, the processing time of Policy Contract (PC), Access Contract (AC) and Device Contract (DC) under a different number of concurrent requests is tested.	The system can store the Uniform Resource Location (URL) of resource data generated by the device and a query method, allow administrator users to manage Attribute Based Access Control (ABAC) policies, and enable user access control of the data.

(Continued)

Table 11 (continued)

Scheme	Technology/Application background	Access control policy	Security	Efficiency	Function
Our scheme	Based on the ACE authorization framework of OAuth 2.0, based on the Hyperledger Fabric platform, based on smart contract technology. The system can be applied to sharing education archives and access control scenarios.	The system adopts a dual-mode access control mechanism combining the DAC and MAC models and improves the access authorization framework based on the ACE authorization framework of OAuth 2.0.	The security of multiple objects has been analyzed in detail, as described in the previous subsection.	Test contract uploading blockchain efficiency, Application Programming Interface (API) operation efficiency, and access control process time indicate that the system solution is within acceptable runtime. The system has modularized the workflow, with different modules run by various system entities, and the overall system efficiency is high.	The system framework can be used with most homomorphic cryptographic retrieval schemes. The process of encryption, storage, retrieval, sharing, and access control of education archives are among multiple education institutions. A network-wide vote conducts user registration. All system entities' action logs are recorded in a tamper-proof manner on the chain.

As can be seen from the Table 11, compared with the centralized trusted institutions used in the literature [18] to manage access permissions, this paper uses a distributed network of blockchain without trust relationship to manage access permissions, which solves the problems of single point failure and third-party service trust; compared with the literature [19] based on static access permissions verification, this paper adopts a combination of static and dynamic access permissions verification mode, which has higher system flexibility and usability; compared with the literature [22,23], which use encryption or signature technology to control access to sensitive data, the security encryption technology used in this paper does not play a role in access control, but ensures the security of data sharing on the chain and reduce security threats from members of the system; compared with the public platform of the Ethernet blockchain used in literature [24], the newly registered system entities in this paper need to be joined after a whole network voting vote; literature [25] is similar to the research scheme of this paper, but the access control mechanism and specific application scenarios adopted by both are different, and this paper also combines homomorphic ciphertext retrieval mechanism and improves the access authorization framework. By comparing various research works, the safety, efficiency, and functional advantages of the scheme in this paper are outstanding, and it has high research significance and application value.

3) Test results

a) Test contract efficiency

On the Hyperledger Fabric, test the running time of installing and instantiating four EduASAC system smart contracts, and test the impact of the average memory of the contract Docker container and the number of APIs in the contract on the contract initialization efficiency, as shown in Fig. 3.

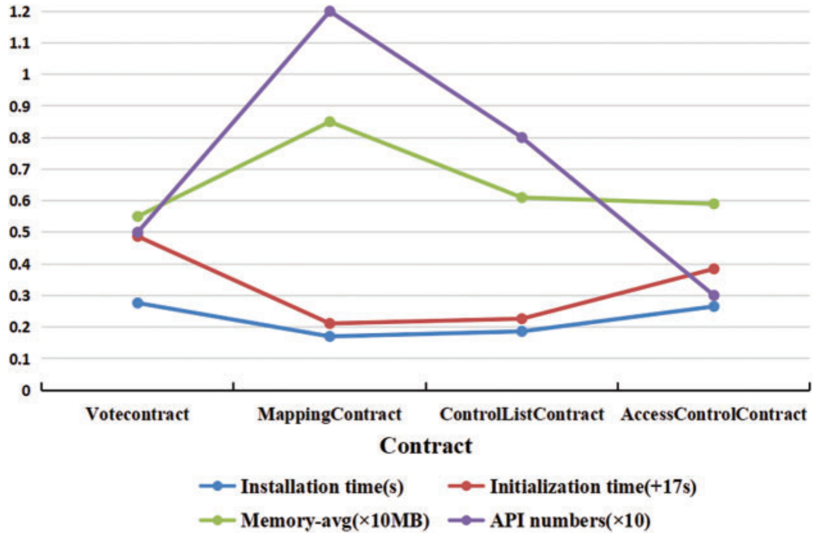

Figure 3: Comparison chart of various factors affecting contract efficiency

Smart contract code written in the Go language generally cannot run directly on the blockchain but needs to be run in a specific sandbox environment Docker container. The installation process is that the smart contract is uploaded to the chain, and the running time is short. The instantiation process needs to call the Docker container and initialize the smart contract, which takes a long time. As can be seen from the figure above, the on-chain efficiency of different smart contracts is related to the amount of memory in the Docker container, the number of APIs, and the complexity of the system behavior controlled by the contract. Tested the average latency and Central Processing Unit (CPU) of the three contracts when writing and querying ledger data, as shown in Fig. 4; tested the average latency and throughput of AccessControlContract under different transaction numbers, as shown in Fig. 5.

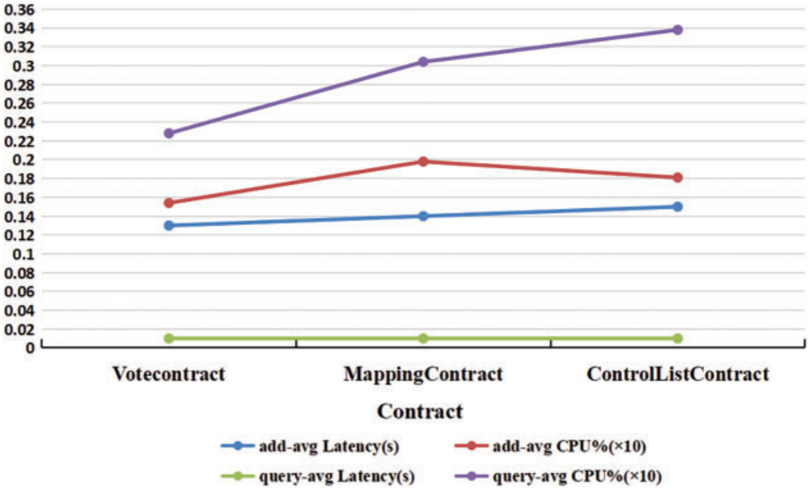

Figure 4: Average latency and CPU when contracts write and query ledger data

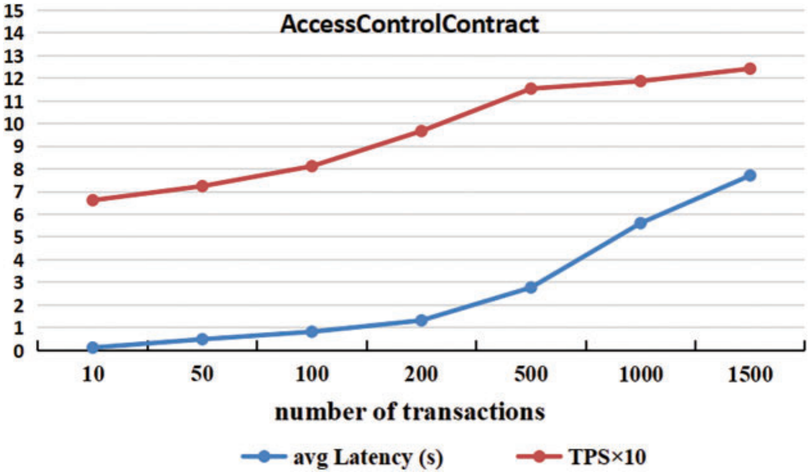

Figure 5: The average latency and throughput of AccessControlContract under different transaction numbers

The experimental results in the above figure show that the EduASAC system can maintain high throughput in a large-scale transaction environment with high efficiency and low latency. It enables all network parties to reach a low-cost consensus to ensure data consistency.

b) Test contract API

To illustrate the system performance of EduASAC, four contracts of the EduASAC system are called, and the running time of each contract API was recorded in detail. The specific test results are shown in Figs. 6–9.

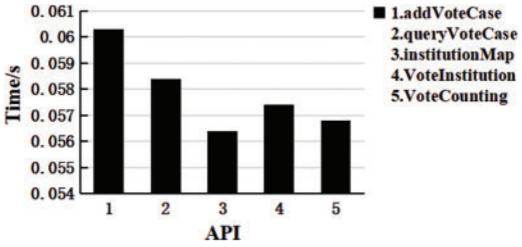

Figure 6: API runtime of Vote Contract

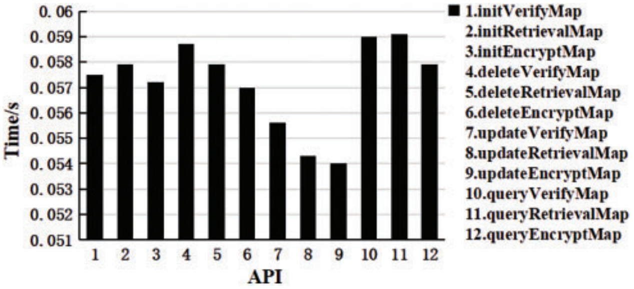

Figure 7: API runtime of Map Contract

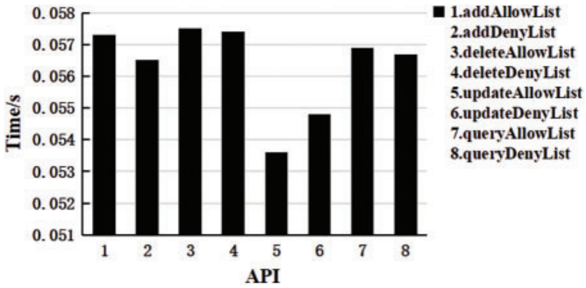

Figure 8: API runtime of Control List Contract

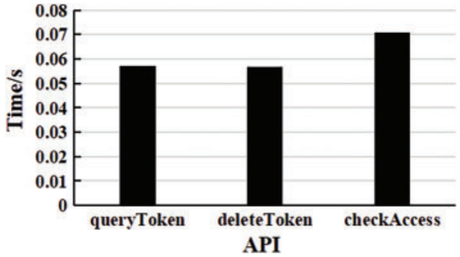

Figure 9: API runtime of Access Control Contract

Note: *checkAccess*() in the Access Control Contract is when VP is set to 0, the API running time without access control.

API is the behavior code for smart contracts to perform specific operations on data on the chain, including adding, deleting, updating, and querying data. As can be seen from the figure above, the operating efficiency of codes of the same behavior is roughly the same, and the subtle difference is the amount of data stored on the chain and the amount of data of the operation object. The scheme in this paper stores and shares the parameters relating to the homomorphic ciphertext retrieval scheme and access control strategy on the chain. Asymmetric algorithms encrypt all of them. Mechanisms with a large amount of calculation are all run off-chain, and education archives with a large amount of data are stored in the storage server. The data operated by smart contracts on the chain has high security, a small amount of data, and simple operation, improving the EduASAC system's operating efficiency to a certain extent.

c) Test access verification

When DO sets different VP parameters for shared data, different access control mechanisms will operate in the Access Control Contract. In the experimental system of EduASAC, the process of DU calling *checkAccess*() in the Access Control Contract to verify the access permission has been simulated. The specific running time of 9 verification access permission workflows has been tested. Table 12 below shows the previous conditions for operating the nine verification access permissions processes.

Table 12: Operating conditions

Number	Operating conditions
1	VP is set to 0, that is, there is no verification access process, only the Token generation part.
2	VP is set to 1, and the S_{Info} of DU is higher than the S_{data} of the target data, allowing DU to access the data.
3	VP is set to 1, and the S_{Info} of DU is lower than the S_{data} of target data, refusing DU to access the data.
4	VP is set to 2, and the system queries the DU information in DAAL, allowing DU to access the data.
5	VP is set to 2, and the system does not query the DU information in DAAL, refusing DU to access the data.
6	VP is set to 3. Although the S_{Info} of DU is higher than the S_{data} of target data, the system queries DU information in DDAL, refusing DU access to the data.
7	VP is set to 3, and the S_{Info} of DU is higher than the S_{data} of target data, the system does not query DU information in DDAL, allowing DU access to the data.
8	VP is set to 3. Although the S_{Info} of DU is lower than the S_{data} of target data, the system queries DU information in DAAL, allowing DU access to the data.
9	VP is set to 3. Although the S_{Info} of DU is lower than the S_{data} of target data, the system does not query DU information in DAAL, refusing DU access to the data.

In the simulation test, if DU's access request to the data is allowed, the test content will include the generation of the Token and the storage process on the chain; otherwise, the contract will end directly after the verification permissions fail. Fig. 10 below shows the running time results of *checkAccess*() in the Access Control Contract under different conditions.

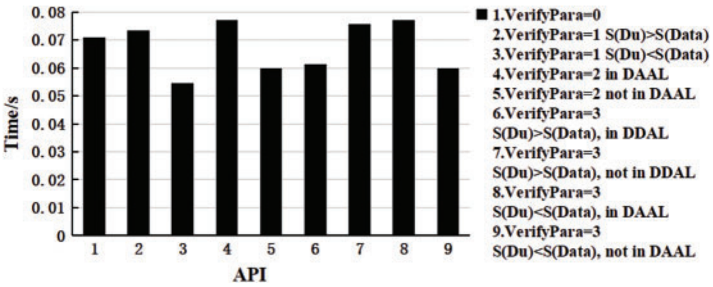

Figure 10: Runtime of checkAccess()

It can be seen from the figure that compared with the running time of *checkAccess*() without access control when VP is 0, the efficiency of the various access control processes designed in this paper is higher, and the running time of Token generation and on-chain storage is shorter. It shows that the EduASAC system using smart contracts for on-chain access control is well-designed, with low operating costs, high efficiency, and high application value.

6 Conclusion and Future Works

The EduASAC system studied in this article takes advantage of the decentralization, tamper-proof, and traceability of blockchain technology, and applies blockchain smart contract technology to implement a dual-mode access control mechanism and a new access authorization framework. At the same time, the system can be combined with most homomorphic ciphertext retrieval schemes running on storage servers to solve the problem of low data security in the storage and sharing of education archives. The paper designs the system workflow of educational archive sharing and access control in detail, including the core modules of the system—access control and homomorphic ciphertext retrieval; introduces the algorithm input, operating logic, and data output of four smart contracts running on the blockchain, emphasizing the role of each smart contract in the system; tests the operating efficiency and memory of the smart contract, as well as the running time of the smart contract API and access control process to prove the availability of the system.

In summary, the design and implementation of the EduASAC system provide a practical reference for other researchers to carry out relevant research. In future work, we can improve in the following areas:

1) The experiments in this paper were conducted on the test network of Hyperledger Fabric. In the future, we will consider deploying a Blockchain as a Service (BaaS) platform closer to the actual blockchain application and further verify the system's performance.
2) This scheme involves the design and implementation of multiple smart contracts. In the future, we will consider using a more professional platform to deploy, schedule and manage smart contracts.
3) Sharing homomorphic ciphertext data involves the security of sharing encryption keys. This paper uses a re-encryption process that allows education archives in a ciphertext state to be shared among education institutions without revealing the encryption key to third-party service providers. In the future, we can try to combine the homomorphic ciphertext retrieval mechanism with proxy re-encryption to jointly complete the secure retrieval and sharing process of data.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Fundamental Research Funds for the Central Universities. Nos. 3282023017, 328202251. RL H received the grant.

Author Contributions: Study conception and design: Ronglei Hu, Chuce He; Data collection: Yaping Chi, Xiaohong Fan; Analysis and interpretation of results: Xiaoyi Duan, Ping Xu; Draft manuscript preparation: Chuce He, Wenbin Gao. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The scheme design, flow, smart contract pseudo-code and related data of this paper have been provided in the paper, readers can reproduce the scheme according to the content of the article. If you need the smart contract code and simulation data, please contact the corresponding author.

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

References

[1] D. Schaffnauser, *Blockchain: Letting Student Own Their Credentials*, Campus Technology, 2017. [Online]. Available: https://campustechnology.com/articles/2017/03/23/blockchain-letting-students-own-their-credentials.aspx (accessed on 30/10/2023)

[2] M. U. Bokhari, Q. Makki and Y. K. Tamandani, "A survey on cloud computing," *Big Data Analytics, Advances in Intelligent Systems and Computing*, vol. 654, pp. 149–164, 2018.

[3] C. Cachin, I. Keidar and A. Shraer, "Trusting the cloud," *ACM Sigact News*, vol. 40, no. 2, pp. 81, 2009.

[4] X. Chen, D. Zou, G. Cheng, H. Xie and M. Jone, "Blockchain in smart education: Contributors, collaborations, applications and research topics," *Education and Information Technologies*, vol. 28, pp. 4597–4627, 2023.

[5] A. O. J. Kwok and H. Treiblmaier, "No one left behind in education: Blockchain-based transformation and its potential for social inclusion," *Asia Pacific Education Review*, vol. 23, no. 3, pp. 445–455, 2022.

[6] R. J. Maestre, J. B. Higuera, N. G. Gómez, J. R. B. Higuera and J. A. S. Montalvo, "The application of blockchain algorithms to the management of education certificates," *Evolutionary Intelligence*, vol. 16, pp. 1967–1984, 2023.

[7] H. Li and D. Han, "EduRSS: A blockchain-based educational records secure storage and sharing scheme," *IEEE Access*, vol. 7, pp. 179273–179289, 2019.

[8] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record," *Adaptive and Adaptable Learning, EC-TEL 2016: Adaptive and Adaptable Learning*, pp. 490–496, 2016.

[9] D. Tapscott and A. Tapscott, "The blockchain revolution and higher education," *Educause Review*, vol. 52, no. 2, pp. 11–24, 2017.

[10] R. A. Mishra, A. Anshuman, A. Braeken and M. Liyanage, "Privacy protected blockchain based architecture and implementation for sharing of students' credentials," *Information Processing and Management*, vol. 58, no. 3, pp. 102512, 2021.

[11] T. Hewa, M. Ylianttila and M. Liyanage, "Survey on blockchain based smart contracts: Applications, opportunities and challenges," *Journal of Network and Computer Applications*, vol. 177, pp. 102857, 2021.

[12] R. Arenas and P. Fernandez, "CredenceLedger: A permissioned blockchain for verifiable academic credentials," in *2018 IEEE Int. Conf. on Engineering, Technology and Innovation (ICE/ITMC)*, Stuttgart, Germany, pp. 1–6, 2018.

[13] P. Ocheja, B. Flanagan, H. Ueda and H. Ogata, "Managing lifelong learning records through blockchain," *Research and Practice in Technology Enhanced Learning*, vol. 14, pp. 1–19, 2019.

[14] F. Hua, Y. Ding, J. Sun, J. Li and W. Shen, "A blockchain-based trusted education and degree certification system," *Cyberspace Security*, vol. 11, no. 9, pp. 9–18, 2020.

[15] W. Yuan, B. Yan, W. Li, L. Hao and H. Yang, "Blockchain-based medical health record access control scheme with efficient protection mechanism and patient control," *Multimedia Tools and Applications*, vol. 82, pp. 16279–16300, 2022.

[16] P. Bagga, A. K. Das, V. Chamola and M. Guizani, "Blockchain-envisioned access control for Internet of Things applications: A comprehensive survey and future directions," *Telecommunication Systems*, vol. 81, pp. 125–173, 2022.

[17] L. Zhang, Y. Zhang, Q. Wu, Y. Mu and F. Rezaeibagha, "A secure and efficient decentralized access control scheme based on blockchain for vehicular social networks," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17938–17952, 2022.

[18] H. Shafagh, L. Burkhalter, A. Hithnawi and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," in *Proc. of the 2017 on Cloud Computing Security Workshop*, Dallas, USA, pp. 45–50, 2017.

[19] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[20] O. Alphand, M. Amoretti, T. Claeys, S. D. Asta, A. Duda *et al.,* "IoTChain: A blockchain security architecture for the Internet of Things," in *2018 IEEE Wireless Communications and Networking Conf. (WCNC)*, Barcelona, Spain, pp. 1–6, 2018.

[21] S. Erdtman, L. Seitz, E. Wahlstroem, G. Selander and H. Tschofenig, *Authentication and Authorization for Constrained Environments (ACE)*, 2017. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-ace-oauth-authz-07 (accessed on 30/10/2023)

[22] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du *et al.,* "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.

[23] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, pp. 44, 2017.

[24] S. Wang, X. Wang and Y. Zhang, "A secure cloud storage framework with access control based on blockchain," *IEEE Access*, vol. 7, pp. 112713–112725, 2019.

[25] H. Liu, D. Han and D. Li, "Fabric-IoT: A blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.

[26] D. Hardt, "The OAuth 2.0 authorization framework," *Internet Requests for Comments*, vol. 6749, pp. 1–76, 2012.

[27] R. L. Rivest, L. M. Adleman and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, pp. 169–179, 1978.

[28] D. Boneh, G. D. Crescenzo, R. M. Ostrovsky and G. Persiano, "Public key encryption with keyword search," in *Int. Conf. on the Theory and Application of Cryptographic Techniques*, Interlaken, Switzerland, vol. 3027, pp. 506–522, 2004.

[29] Y. Yang, Y. Zhao, J. Zhang, J. Huang and Y. Gao, "Recent development of theory and application on homomorphic encryption," *Journal of Electronics and Information Technology*, vol. 43, no. 2, pp. 475–487, 2020.