International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 8172-8187

https://doi.org/10.70517/ijhsa464700

Blockchain-based collaborative mechanism for cross-institutional education data sharing

Wei Yue[1,*], Yufeng Zhou[2] and Yongtao Nie[1]
[1] Innovation and Entrepreneurship Guidance Center, Weifang Engineering Vocational College, Weifang, Shandong, 262500, China
[2] School of Marxism, Weifang Engineering Vocational College, Weifang, Shandong, 262500, China
Corresponding authors: (e-mail: 19506533327@163.com).

Abstract Blockchain technology provides a decentralized, tamper-proof solution for cross-institutional education evaluation data sharing. This study proposes a cross-institutional education data sharing model that integrates blockchain, RSA cryptographic accumulators, and IPFS to achieve secure off-chain storage and efficient on-chain verification of education data. Using the RSA accumulator, multiple fingerprints of educational record data are aggregated into a single cryptographic accumulator for on-chain storage, enabling users to quickly verify the authenticity of individual data points via verifiable credentials. Attribute-based encryption (CP-ABE) is employed to protect the original data stored in IPFS, with students able to define their own access policies to ensure granular permission control. The experiment validated performance using the real-world dataset MOOCCube. RSA accumulator key slicing processed 200 key pairs in just 62.69 seconds, improving efficiency by 14.5% compared to the Slicing method. Hybrid encryption of 200 courses took 99.43 seconds, and smart contract management of 50 contracts took only 247.16 seconds, both significantly outperforming comparison schemes. Combined with Bloom filters to enable multi-keyword search, a 5-keyword search takes only 4.26 seconds, which is 67.1%–114.6% faster than the baseline scheme. A group-optimized consensus mechanism is designed to improve throughput, reaching a peak of 1023.61 TPS, which is 3.6 times higher than the ordinary scheme. Block recovery success rate reaches 100% when the replication factor c ≥ 2, and the direct recovery rate remains at 89.53% even when the node scale is expanded to 40 nodes. This model effectively improves the efficiency and scalability of cross-institutional education data sharing while ensuring data privacy and integrity.

Index Terms blockchain, education data sharing, RSA, cross-institutional collaboration, data privacy

I. Introduction

During educational activities, a vast amount of educational data is generated, including teaching behavior data, learning process data, digital educational resources, and teacher-student archive data, among others. This data and information holds immense value, serving as a continuous driving force for innovation and development in the education sector and providing crucial scientific support for educational reforms [1]-[3]. However, some educational data resources involve teacher-student privacy, such as educational archive data. If leaked during sharing, such data could lead to numerous security issues [4], [5]. To ensure that these educational data resources can be fully utilized, truly promote the sharing of educational resources, and enhance resource utilization efficiency, the education industry is actively exploring new avenues for educational informatization.

With the rapid advancement of IT technology, data-driven methods are widely adopted to enhance the efficiency of educational data storage and sharing [6]. While current educational data resource sharing systems have brought significant conveniences to data management, several core issues remain to be addressed:

(1) Compared to traditional paper-based data resources, digital resources are stored in the form of bits and bytes, making them more susceptible to alteration. They are also more prone to tampering during storage, transmission, and processing, which is a major issue currently faced [7].

(2) Different educational institutions operate as “islands” of educational resources, lacking secure and effective channels for sharing educational data resources [8].

(3) Existing educational data resource protection schemes are mostly based on centralized storage solutions, which have poor security. Once data resources are tampered with or destroyed, they are generally difficult to recover, and the privacy of archival data is not protected [9].

Addressing issues such as the forgery or tampering of educational data, privacy leaks, and data silos caused by centralized storage in cross-institutional educational data management, the emergence of blockchain technology has introduced a new solution to data integrity and tamper-proofing issues [10]-[12]. As a cutting-edge technology,

blockchain is renowned for its decentralized and tamper-proof characteristics [13]. Within a blockchain network, nodes collectively maintain a distributed ledger based on a unique consensus mechanism, ensuring the authenticity and integrity of data [14]. The characteristics of multi-party consensus, decentralization, tamper-proofing, and programmability provide technical support for educational data sharing, making it an effective method for promoting fair, secure, and efficient inter-institutional educational data sharing [15]-[17].

Since 2016, the frequency of research keywords related to “blockchain” data sharing has shown a trend of annual growth, with primary directions including educational data sharing, learning achievement certification, medical data storage and sharing, cloud storage data integrity verification, and identity authentication [18]-[21]. For example, Li, H, and Han, D [22] utilized blockchain technology, storage servers, and encryption techniques to achieve secure storage and sharing of educational records across institutions, ensuring their reliability and security. Through preliminary testing, they validated the effectiveness of the “Educational Blockchain Storage Sharing System (EduRSS).” Tanriverdı, M [23] proposed a PublicEduChain framework based on blockchain technology, which transfers control of educational data to students and utilizes public blockchain networks to ensure data security, decentralized management, and sharing. Li, Z, and Ma, Z [24] proposed a blockchain-based solution for the secure storage and sharing of educational record data, which utilizes consortium blockchain, smart contracts, and encryption technology to ensure privacy, efficiency, and stability.

In the data sharing process, there are two main methods of data notarization: hash notarization and Merkle Root notarization. Abdul Hadi, Z, and Au, T [25] proposed uploading all certificates of Vietnamese high school and higher education students to the TomoChain public blockchain in Singapore via hash notarization, ensuring that these records are both transparent and tamper-proof. This initiative aims to build a more transparent and tamper-proof record system. Reza, A et al. [26] utilized Merkle Root evidence to enhance document security, reduce fraud, and shorten identity verification time. They employed consortium blockchain technology to provide a stable and efficient platform for verifying academic records, facilitating inter-university exchanges, and publishing job postings. Balobaid, A et al. [27] proposed a blockchain-based data management system with encryption capabilities for educational institutions to securely manage educational records. They introduced a novel Merkle tree-based strategy and utilized DNA sequences and chaotic systems to enhance security and authentication capabilities. Hameed, B et al. [28] explored the application of blockchain technology in the education sector, studying various projects and protocols to promote its adoption in education, emphasizing its advantages in terms of security and reducing paperwork. As such, blockchain-based education data sharing and privacy protection solutions remain a key area of focus, offering significant practical application value for optimizing education data governance capabilities and building fair, secure, and efficient education data sharing platforms.

This study first constructs the technical framework of the overall sharing model. The model defines four core entities: data owners (DO), data access users (DU), the blockchain network, and IPFS distributed storage. It details the functional roles of each entity, interaction processes, and the secure storage and access control mechanisms for data both on-chain and off-chain. Next, the study systematically reviews the key mathematical principles underlying the operation of the RSA accumulator. It focuses on modular arithmetic and its core properties (such as congruence relations and operational laws), Euler's theorem and its corollaries (Fermat's Little Theorem), and the concept of modular inverses and their existence proofs. The theory and architecture are then specifically applied to the educational evaluation data sharing scenario. The paper also provides a detailed design for the methods of storing educational archive data on the blockchain and verifying educational archive data. The former describes how to encrypt and store raw educational data on IPFS, then use the RSA accumulator algorithm to efficiently aggregate the “fingerprints” of multiple data points into a cryptographic accumulator for storage on the blockchain. The process includes data selection, IPFS storage, accumulator construction, evidence ID generation, and smart contract-triggered on-chain storage. The latter designs a lightweight off-chain verification process. Users submit specific data items for verification, and the system retrieves the corresponding verifiable credentials from the database and retrieves the corresponding accumulator evidence from the chain. Using the RSA accumulator's verification algorithm, only the data item, its credential, and the on-chain evidence are required to quickly verify whether the single data item is correctly included in the original accumulator, without exposing other data or downloading all on-chain information. Additionally, the process briefly explains the attribute-based encryption (CP-ABE) and decryption procedures when data is stored on IPFS, ensuring access control and privacy protection for the original data.

II. Building a cross-institutional education evaluation data sharing model based on blockchain and RSA

II. A.Blockchain-based academic data sharing model

The blockchain-based academic data sharing model architecture includes four main entities: data owners (DO), data access users (DU), blockchain, and IPFS. These entities each perform the following functions:

Data owners (DO) are responsible for effectively managing their academic data resources and have the right to authorize or revoke access permissions for various data access users. Data owners play a key role in data sharing. They are responsible for defining data access policies and can upload academic performance data to the blockchain for better management and sharing. DOs have flexible identities and can also serve as data access users (DUs) to query and access academic performance data.

Data access users (DUs) are users who wish to access academic performance data. Their primary tasks are to perform data upload queries and submit data access requests. DU can only successfully access data when specific access policies are met, through the verification of permission determination contracts. DU identities are multifaceted; they can also assume the role of data owners (DO) and be responsible for publishing and sharing data.

Blockchain technology builds a trusted infrastructure that solves trust issues between participants and provides multiple services, including data upload, retrieval, and sharing. It also accurately records every user operation on the data, ensuring its security and traceability throughout its lifecycle. Before attempting to access data, users must first pass through the blockchain network's initial access control to verify their identity. Subsequently, access control policies classify users into different access permission levels, retrieve the hash values of the corresponding academic data storage, use them as credentials to retrieve the original data from IPFS, and return it to the user.

IPFS provides data owners (DO) with a decentralized, secure storage method. It is responsible for storing the initial data uploaded by DO and creating content identifiers (CID) as references for users to retrieve data. IPFS utilizes redundant backup mechanisms to ensure stable data storage while safeguarding data security and immutability. This process enhances data reliability and accessibility.

By defining different levels of data access permissions based on privacy requirements, the aforementioned process achieves secure storage, data isolation, and controlled management of data at different levels within the chain, while ensuring data reliability and immutability. This enhances data credibility and accessibility while meeting security requirements for data sharing.

II. B.Mathematical Foundations of the RSA Algorithm

To achieve efficient, fine-grained verification of data authenticity, we introduce the RSA algorithm as a key technical component. The effective implementation of the RSA cryptographic accumulator relies on a solid mathematical foundation. The following sections will provide a detailed introduction to the concepts of modular arithmetic, Euler's theorem, and modular inverses in this algorithm.

II. B. 1) Model operations and their rules

For any integer x and positive integer n, there must exist integers k and a such that the equation $x = kn + a$ holds, where $0 \le a \le n$. Then $a = x \bmod n$ or $a = x\%n$ can be called the modulo operation of a on x with respect to n.

For integers x and y and a positive integer n, if $(x-y)/n$ is an integer, then the integers x and y are said to be congruent modulo n, which can be written as $x \equiv y \bmod n$.

The modulo operator has the following properties:

(1) $x \equiv y \bmod n$ is equivalent to $y \equiv x \bmod n$;

(2) $x \bmod n = y \bmod n$ is equivalent to $x \equiv y \bmod n$;

(3) If $x \equiv y \bmod n$ and $y \equiv z \bmod n$, then $x \equiv z \bmod n$.

Similar to arithmetic operations, modulo operations satisfy the commutative law, associative law, and distributive law, with the exception of division. The relevant operational rules are as follows:

$$(x+y) \bmod n = (x \bmod n + y \bmod n) \bmod n \tag{1}$$

$$(x-y) \bmod n = (x \bmod n - y \bmod n) \bmod n \tag{2}$$

$$(x \times y) \bmod n = (x \bmod n \times y \bmod n) \bmod n \tag{3}$$

$$x^y \bmod n = (x \bmod n)^y \bmod n \tag{4}$$

It is known that exponential operations can be viewed as multiple identical multiplication operations. Similarly, formula (4) can convert modular exponentiation into modular multiplication. This formula is frequently used in the RSA algorithm and can greatly optimize the algorithm's computational efficiency.

II. B. 2) Euler's Theorem

In number theory, for a positive integer n, the Euler function $\varphi(n)$ is the number of positive integers less than or equal to n that are coprime to n. This function is named after its first researcher, Euler. It is also called the φ function (named by Gauss) or the Euler totient function (named by Sylvester). The general formula for the Euler function:

$$\varphi(n) = n * \left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\left(1-\frac{1}{p_3}\right)\left(1-\frac{1}{p_4}\right)\cdots\left(1-\frac{1}{p_n}\right) \tag{5}$$

where $p_1, p_2, \cdots, p_n$ are all prime factors of n, and n is a non-zero integer. For example: $8 = 2\times 2\times 2$, $\varphi(8) = 4$ (1, 3, 5, and 7 are all coprime to 8), $15 = 3\times 5$, $\varphi(15) = 8$ (1, 2, 4, 7, 8, 11, 13, and 14 are all coprime to 15).

Euler's theorem studies the properties of congruence. The core of the RSA algorithm is Euler's theorem, which states that if there are two positive integers n and a that are coprime, and $\varphi(n)$ is the Euler function of n, then they satisfy the relationship:

$$a^{\varphi(n)} \equiv 1 \bmod n \tag{6}$$

Fermat's Little Theorem: If p is a prime number, then for any integer a, the following holds:

$$a^{p} \equiv a \bmod p \tag{7}$$

Thus, using Euler's theorem and Fermat's little theorem, we can deduce that if positive integers a, n are coprime, then for any positive integer b, we have:

$$a^{b} \equiv a^{b \bmod \varphi(n)} \bmod n \tag{8}$$

II. B. 3) Model elements

If two positive integers a and n are coprime, then there exists an integer b such that $ab-1$ is divisible by n. In this case, b is called the “modular inverse” of a. For example, 3 and 11 are coprime because $(3\times 4)-1$ is divisible by 11, so the modular inverse of 3 is 4. Clearly, a number can have more than one modular inverse; any integer multiple of 4 plus or minus 11 is a modular inverse of 3. That is, if b is the modular inverse of a, then $b+kn$ is also the modular inverse of a. The existence of modular inverses can be proven using Euler's theorem:

$$a^{\varphi(n)} = a\times a^{\varphi(n)-1} \equiv 1 \bmod n \tag{9}$$

Based on the above formula, we can deduce that the $\varphi(n)-1$ th power of a is the modular inverse of a.

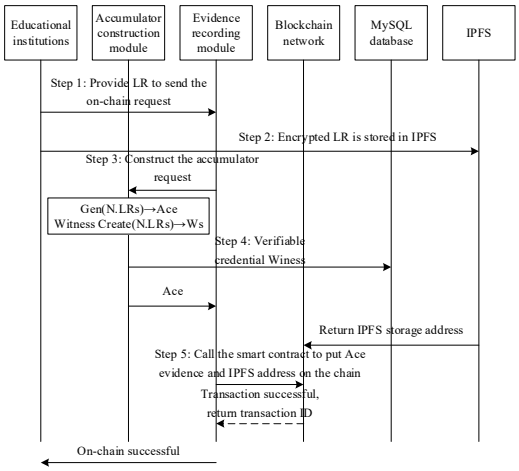

Figure 1: Flowchart of educational archive data on-chain evidence storage

II. C.Education Data Evidence Storage and Verification Method Based on RSA Cryptographic Accumulator

Based on the data evidence model structure proposed above using the RSA cryptographic accumulator, this section elaborates on the evidence and verification methods for educational data using the RSA cryptographic accumulator in the context of educational data sharing applications.

II. C. 1) Method for recording educational archive data on the blockchain

The education archive data chaining algorithm aims to efficiently construct and securely store education archive data on the blockchain to ensure its immutability and traceability. In the education data sharing scenario, the entities involved include educational institutions, accumulator construction modules, chaining modules, consortium chain

networks, MySQL databases, and IPFS distributed databases. The process of chaining education archive data is shown in Figure 1.

The specific steps of the on-chain process are as follows:

The specific steps of the on-chain process are as follows:

(1) Educational institutions select the learning records LR that need to be stored on the chain in batches, and send a request to the chain;

(2) LR is encrypted by the attribute base and uploaded to the distributed database IPFS for file query and authorized access, and the ownership of the file data belongs to the student, and the student sets the attribute access policy;

(3) The password accumulator construction algorithm is used to construct LR into a password accumulator for proof, and the verifiable credential $witness$ is generated. Here's how to construct it:

$$\begin{aligned} &Gen(N, LR_n) \to acc : acc = g^{hash(LR_1),...,hash(LR_n)} \bmod N \\ &WitnessCreate(pk_{acc}, x_i, acc) \to witness : w_i \\ &= g^{hash(LR_1),...,hash(LR_{i-1}),hash(LR_{i+1}),...,hash(LR_n)} \bmod N \end{aligned} \tag{10}$$

Among these, $LR_1,...,LR_n \in LR$, and N is generated by the $Init(\ )$ stage constructed by the accumulator.

The cryptographic accumulator algorithm is a one-way algorithm that can aggregate multiple archive data into a single evidence, thereby hiding each aggregated learning archive data, and can verify each aggregated educational archive information using a verifiable credential $witness$.

(4) Calculate the unique ID that uniquely identifies a cryptographic accumulator evidence. Academic credentials ID = Hash(graduation institution, graduation date); honor credentials ID = Hash(competition name, award date); course transcript credentials ID = Hash(course-offering institution, date, course name).

(5) Trigger the education data evidence smart contract to store the password accumulator evidence on the blockchain. The evidence structure includes the evidence ID, the public key of the institution uploading the evidence, the password accumulator parameters (Key_N, Key_G), and the accumulator evidence Acc. A successful upload will return the transaction id.

II. C. 2) Methods for verifying educational record data

The core of verifying the authenticity of educational record data is to verify whether the educational record data is a member of the on-chain cryptographic accumulator. If so, the educational record information to be verified is authentic; otherwise, it is not authentic.

The verification process for educational record data is shown in Figure 2 below.

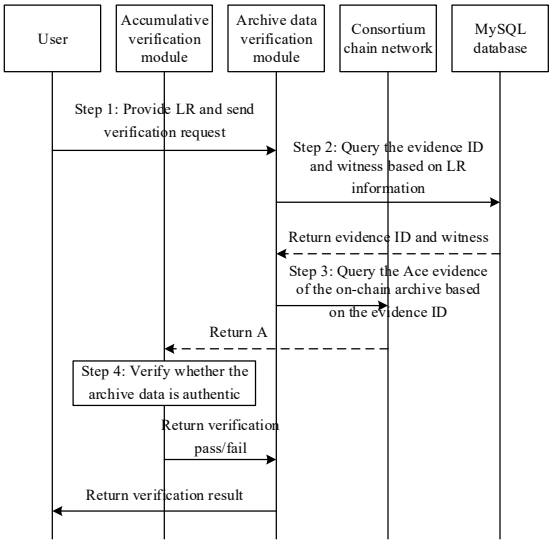

Figure 2: Flowchart for Verifying Educational Archive Data

(1) The user inputs the archive data information to be verified LR. If the archive information to be verified is academic record information, the user must input the following: name, ID number, graduating institution, enrollment date, graduation date, major, graduation/completion status, mode of study, and degree. If the archive information to be verified is competition honor information, the user must input the following: name, ID number, full name of the competition, competition level, and award date. If the archive information to be verified is course grade information,

the user must input the name, ID number, institution where the course was offered, course name, and course offering date.

(2) Query the MySQL database based on the LR information. If no match is found, it indicates that the archive data for that LR has not yet been recorded on the blockchain. Otherwise, return the archive data's recording ID and the verifiable credential $witness$ required for verification.

(3) Retrieve the cryptographic accumulator recording Acc stored on the blockchain based on the archive data's recording ID.

(4) Use the accumulator verification algorithm to verify the user-input educational record data to be verified. The verification method is as follows: $VerMem\left(pk_{acc}, LR_i, witness, acc\right) \rightarrow True / False : acc = witness_i^{hash(LR_i)} \bmod N$.

If the verification passes, return $True$, indicating that the user-inputted educational record data information is authentic; otherwise, if the verification fails, return $False$.

In the educational record data verification based on cryptographic accumulator evidence, the input is the educational record to be verified $verifyLR$, and the output is the verification result $result$.

The encryption process for educational record data $\left(LR \rightarrow encryptLR\right)$: First, the key generation algorithm is invoked to generate a symmetric key key for the educational record information. Then, the AES symmetric encryption algorithm is invoked to perform symmetric encryption on the educational record information LR, resulting in the ciphertext $encryptLR$ of the educational record information LR. At this point, the symmetric key key is encrypted using CP-ABE with the access structure A set by the owner of the educational record to generate $encryptKey$, which is then stored in the server's local file. The access structure A is a logical expression constructed based on attributes and the intrinsic relationships between attributes.

The decryption process of educational record data $\left(encryptLR \rightarrow LR\right)$: The process of decrypting $encryptLR$ to obtain $decryptLR$ involves retrieving the symmetric key ciphertext $\mathrm{encryptKey}$ stored locally on the server, using the attribute private key SK of the educational institution administrator and the public key PK generated by the server during initialization, decrypting $encryptKey$ using CP-ABE to obtain the symmetric key plaintext key, and then decrypt the encrypted educational archive data $encryptLR$ using AES with the key key to obtain the plaintext $decryptLR$ of the educational archive data.

III. RSA blockchain performance experiments and analysis based on the MOOCCube dataset

Based on the blockchain-RSA cross-institutional education data sharing model constructed in Chapter 2, this chapter will rely on the real education dataset MOOCCube to verify the actual performance of the model in terms of data storage efficiency, privacy protection strength, system scalability, and block recovery through multidimensional experiments, providing empirical support for cross-institutional education evaluation data sharing.

III. A. Experimental setup

III. A. 1) Experimental Dataset and Experimental Environment

The experimental dataset is the popular MOOCCube dataset, with a size of 4.2GB. The MOOCCube dataset includes over 700 real online courses, approximately 40,000 instructional videos, and tens of thousands of course selection records and course video viewing records generated by nearly 20,000 real users of the MOOC platform. Each course dataset is in the MB range, and the course data includes course ID, course name, course requirements, course details, and other information.

The experimental environment consists of an Ubuntu 24.04 LTS operating system, an NVIDIA RTX 3090 graphics card, 16 GB DDR5 memory, an Intel Xeon Gold 6248R processor, a 2 TB NVMe hard drive, the Visual Studio Code compiler, Web3.py V6.14.0, and Python 3.8.10.

In the experiment, the algorithm based on RSA cryptography and Shamir secret sharing for key splitting and improved sharing phase was compared with the following algorithms: (1) the Normal method, which does not use key splitting or data-layered access control and is managed by a regular smart contract; (2) the Slicing method, which splits the key into segments of a certain number of bits; (3) the SM2 method using the SM2 encryption algorithm, and (4) the AES method using the AES encryption algorithm.

III. A. 2) Evaluation Criteria

The experimental evaluation criteria are the processing times for different stages of the algorithm, with the following calculation formulas

$$t_{PTT} = \sum_{i=1}^{n} TransS\left(K_{PubKey,i}, K_{PKey,i}\right) \tag{11}$$

$$t_{PTH} = \sum_{i=1}^{m} HybirdEncryption\left(D_i\right) \tag{12}$$

$$t_{PTS} = \sum_{i=1}^{l} SCManagement\left(L_{SCList,i}\right) \tag{13}$$

In Equations (11)–(13), tPTT denotes the processing time for the key splitting phase; tPTH denotes the processing time for the hybrid encryption phase; tPTS denotes the processing time for the smart contract management phase; KPubKey,i and KPKey,i denote the key pairs requiring improved key splitting for Shamir secret sharing in the i-th group; Di denotes the plaintext of the i-th group requiring hybrid encryption; LSCList,i denotes the set of smart contracts to be executed for the i-th group; TransS denotes the key cutting method; HybridEncryption denotes the hybrid encryption method; SCManagement denotes the smart contract management method; n denotes the total number of key pairs requiring key cutting; m denotes the total number of data items requiring hybrid encryption; l denotes the total number of smart contracts.

III. B. Comparison of processing times for key cutting, encryption, and contract management

This section focuses on verifying the efficiency of the model's core components. First, it compares the processing times of the key cutting stage, the mixed encryption stage, and the smart contract management stage, and analyzes the performance advantages of the RSA accumulator in hierarchical data management. In the time evaluation experiment, each stage of the operation was repeated 10 times, and the final experimental results were analyzed based on the average processing time of the 10 evaluation experiments.

III. B. 1) Comparison of processing times during the key cutting phase

The processing times for different key cutting methods are compared in Table 1. To more clearly show the comparison of processing times between different methods during the key cutting stage, a line graph is plotted as shown in Figure 3.

Table 1: Comparison of processing times for different key splitting methods

The number of key pairs for key sharing	RSA	Normal	Slicing	SM2	AES
50	13.19	10.27	22.65	16.45	17.19
100	33.94	34.30	42.31	36.11	39.03
150	45.95	44.84	53.57	55.41	57.58
200	62.69	65.23	73.34	67.78	73.96

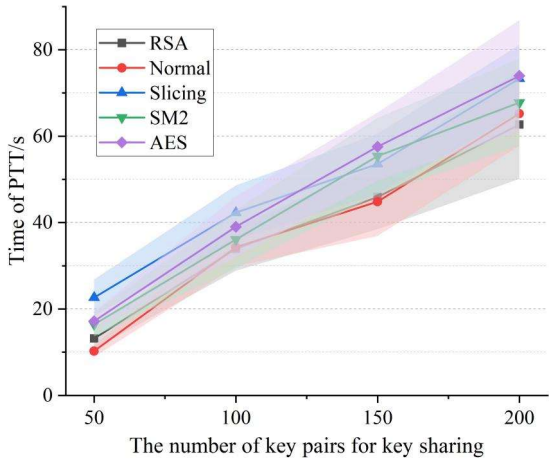

Figure 3: Comparison of processing times for different key splitting methods

The RSA method took 13.19 seconds with 50 key pairs, slightly higher than the Normal method's 10.27 seconds, but significantly lower than the Slicing method's 22.65 seconds. As the number of keys increases to 200 pairs, the RSA method takes 62.69 seconds, far below the Slicing method's 73.34 seconds, SM2's 67.78 seconds, and AES's 73.96 seconds, and only slightly higher than the Normal method's 65.23 seconds.

RSA's processing time is consistently lower than that of the Slicing, SM2, and AES methods, particularly at 50 key pairs, where it is 41.8% faster than Slicing (9.46 seconds); Compared to Normal, RSA incurs additional overhead (28.4% higher for 50 key pairs) due to the introduction of Shamir secret sharing key splitting, but the efficiency gap narrows as the number of keys increases (only 3.9% lower for 200 key pairs). The reason RSA's processing time is higher than the Normal method is that the Normal method does not perform key slicing using Shamir secret sharing, while RSA performs key slicing using the Shamir secret sharing key slicing method, hence RSA's processing time is slightly higher than the Normal method. The reason why RSA's processing time is lower than the Slicing method is that the Slicing method divides the key into fixed-length segments and transmits the key segments to participating nodes separately, resulting in higher transmission latency. In contrast, RSA divides the key using the Shamir secret sharing key splitting method and transmits the key segments simultaneously to participating nodes, resulting in lower key transmission latency. Therefore, the processing time of the RSA method is lower than that of the Slicing method. Since the RSA method uses an improved sharing algorithm based on Shamir secret sharing for key splitting, it avoids transmitting key fragments in plaintext form. Compared to other methods, RSA achieves more efficient and secure encryption effects.

III. B. 2) Comparison of processing times in the mixed encryption phase

The processing times for different mixed encryption methods are compared in Table 2 and Figure 4.

Table 2: Comparison of processing times for different hybrid encryption methods

The number of courses using hybrid encryption	RSA	Normal	Slicing	SM2	AES
50	25.29	33.29	30.10	35.55	31.93
100	54.87	65.14	58.77	61.52	61.04
150	75.34	85.16	79.25	83.35	86.52
200	99.43	107.44	96.07	104.28	113.38

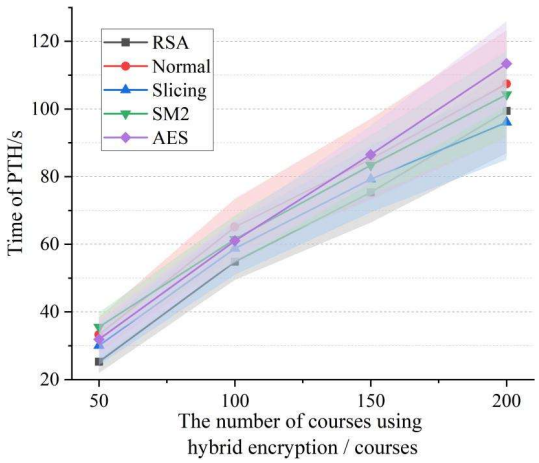

Figure 4: Comparison of processing times for different hybrid encryption methods

The RSA method takes only 25.29 seconds to encrypt 50 courses, significantly faster than Normal's 33.29 seconds, SM2's 35.55 seconds, and AES's 31.93 seconds. When the number of courses increases to 200, RSA's 99.43 seconds remains the fastest, 7.4% faster than Normal's 8.01 seconds and 12.3% faster than AES's 13.95 seconds.

RSA is the fastest method across all data scales. For every additional 50 courses, RSA's processing time increases by an average of 24.7 seconds, a growth rate lower than Normal's 24.8 seconds and AES's 27.5 seconds, indicating superior scalability.

III. B. 3) Comparison of processing times during the smart contract management phase

The processing times for different smart contract management methods are compared in Table 3 and Figure 5.

Table 3: Comparison of times for different smart contract management methods

The number of executed smart contracts	RSA	Normal	Slicing	SM2	AES
10	38.60	77.71	59.29	57.71	70.38
20	97.50	140.27	115.02	107.00	116.60
30	157.98	206.68	170.74	161.14	172.33
40	194.60	254.39	205.78	205.78	215.29
50	247.16	337.73	264.68	280.61	294.97

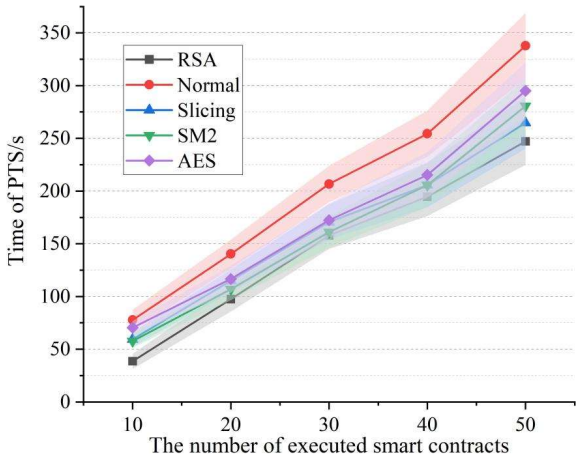

Figure 5: Comparison of times for different smart contract management methods

The RSA method demonstrates the most significant advantages, completing 10 contracts in just 38.6 seconds—50.3% faster than the Normal method's 77.71 seconds and 34.9% faster than the Slicing method's 59.29 seconds. When the number of contracts reaches 50, RSA's 247.16 seconds is still 26.8% faster than Normal's 90.57 seconds and 11.9% faster than SM2's 33.45 seconds.

It can be seen that RSA's processing time is consistently lower than all comparison methods, and the advantage becomes more pronounced as the scale increases (saving 90.57 seconds compared to Normal at 50 contracts). For every additional 10 contracts, RSA's processing time increases by an average of 52.39 seconds, a growth rate significantly lower than Normal's 65.01 seconds and AES's 56.15 seconds, validating the robustness of its management efficiency.

III. C. Encrypted Search and Multi-Keyword Retrieval Performance Verification

In addition to basic operational efficiency, educational data sharing scenarios have higher requirements for secure retrieval. This section further tests the stability of the encrypted search scheme based on Bloom filters under dynamic attribute growth and verifies the applicability of multi-keyword retrieval in actual educational queries.

III. C. 1) Search Performance Comparison

Table 4 compares the performance of the proposed scheme with the aforementioned four search algorithms. The experiment randomly selected 100 educational record files from the MOOCCube dataset to test the search performance of the five algorithms. To accurately reflect the efficiency of the search algorithms, this experiment does not consider the impact of the consensus mechanism. The other four search schemes are all encrypted data search schemes based on ABE access control. The Normal and Slicing schemes completely outsource the search process to cloud servers, while the SM2 and AES schemes use blockchain to assist in encrypted search. However, regardless of the design, both require attribute-based encryption permission verification during the search process.

Unlike the baseline scheme, the attribute-based encryption based on the RSA scheme designed in this paper is applied to the second-stage symmetric key encryption. The search algorithm overhead in this scheme primarily stems from the linear algebra operations of two Bloom filters—the secure index and the encrypted query keyword.

Table 4: Search performance of the 5 schemes in cost of time

Number of attributes	RSA	Normal	Slicing	SM2	AES
0	0.329	0.219	0.186	0.143	0.132
10	0.329	0.527	0.373	0.450	0.406
20	0.307	0.955	0.736	0.824	0.791
30	0.450	1.648	1.472	1.395	1.406
40	0.285	2.164	2.043	1.791	1.879
50	0.384	2.780	2.516	2.153	2.253

To more clearly illustrate the differences in search times between the various methods, a bar chart comparing the performance of the five methods is shown in Figure 6.

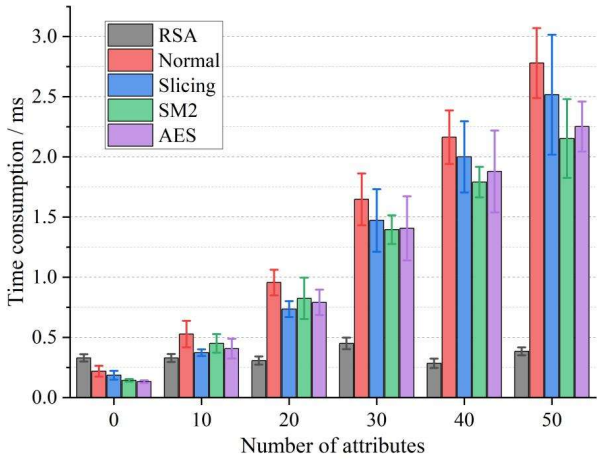

Figure 6: Search performance of the five schemes in cost of time

The search time for the RSA scheme remains relatively stable at 0.285–0.450 seconds, unaffected by an increase in the number of attributes. When the number of attributes increases from 0 to 50, the elapsed time fluctuates by only 0.045 seconds (0.384 seconds vs. 0.329 seconds). In contrast, the search time for the comparison schemes (Normal/Slicing/SM2/AES) increases significantly with the number of attributes. When the number of attributes is 0, all schemes are efficient (AES is the fastest: 0.132 seconds); when the number of attributes reaches 50, the Normal scheme's search time surges to 2.78 seconds (an increase of 1168%), while RSA remains at 0.384 seconds.

Through the experiment, it can be seen that when the number of attributes is 0, this scheme incurs higher algorithmic overhead than the other four schemes due to the need for vector dot product operations during the search. As the number of attributes increases, the search algorithmic overhead of the comparison schemes grows increasingly larger. Other schemes, which rely on attribute encryption permission verification, exhibit a positive correlation between search time and the number of attributes (e.g., the Normal scheme's search time increases by an average of 0.64 seconds for every 10 additional attributes). The RSA algorithm remains stable, being 82.3% faster than the fastest SM2 scheme when the number of attributes is 50 (0.384 seconds vs. 2.153 seconds). It achieves constant low latency through Bloom filtering and vector operations, demonstrating a significant efficiency advantage.

III. C. 2) Multi-keyword search efficiency

The multi-keyword search efficiency experiment was conducted by randomly selecting 100 educational record files for testing. To accurately reflect the efficiency of the search algorithm, the influence of the consensus mechanism was not considered. This paper's approach uses Bloom filters and independent hash functions to convert encrypted file indexes and query keywords into vectors. If two vectors have the same keyword, the corresponding position values are 1, so the query keyword can be determined to be in the encrypted index through a simple vector inner product. This design achieves efficient multi-keyword search functionality. Compared with the baseline approach, which only supports single-keyword search, The experimental results are shown in Table 5 and Figure 7.

Table 5: Experiment on the Cost of Multi-keyword Search

Key word count	RSA	Normal	Slicing	SM2	AES
1	2.46	1.18	1.32	1.98	1.63
2	3.03	3.60	3.21	3.56	3.07
3	3.38	4.92	4.26	3.95	4.04
4	3.91	7.47	7.16	5.71	6.29
5	4.26	9.14	8.83	7.12	7.69
6	4.66	11.12	10.64	8.70	9.62
7	5.01	13.06	12.40	10.02	11.03

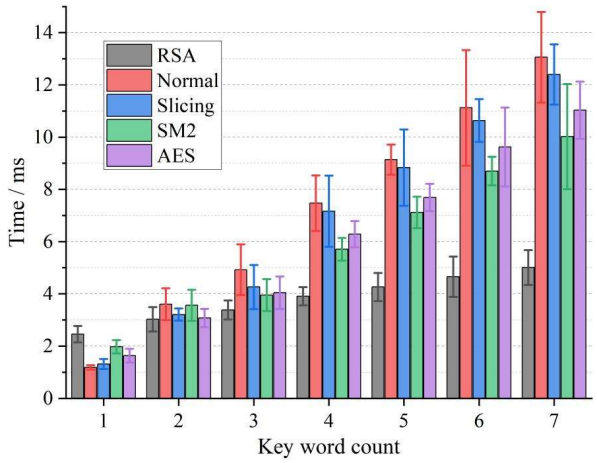

Figure 7: Experiment on the Cost of Multi-keyword Search

From the experiment, it can be analyzed that, under single-keyword retrieval conditions, the efficiency of this scheme is inferior to that of the two comparison schemes. The RSA scheme is the slowest (2.46s), which is 108.5% slower than the Normal scheme's 1.18s. This is because the retrieval process of this scheme requires vector inner product calculations, while the other schemes only require character matching; When the number of keywords is 2, since this scheme supports multi-keyword search functionality, its search efficiency outperforms the other schemes as the number of keywords increases, with RSA efficiency surpassing the others: when there are 2 keywords, it takes 3.03 seconds, which is lower than Normal's 3.60 seconds and AES's 3.07 seconds; When the number of keywords is 5, RSA's 4.26 seconds is 114.6% faster than Normal's 9.14 seconds and 67.1% faster than SM2's 7.12 seconds. Clearly, as the number of keywords increases, the search efficiency advantage of this scheme becomes increasingly evident. In practical educational data search scenarios, support for multi-keyword search is common and necessary, making this scheme more suitable for multi-keyword search in educational scenarios.

III. D. Load testing of consensus algorithm throughput and communication overhead

Given that cross-institutional collaboration requires high-concurrency consensus support, this section will evaluate the network communication efficiency of the model when scaling node size: through throughput (TPS) and communication frequency tests, we will reveal the improvement effects of packet optimization mechanisms on the load of large-scale educational data sharing networks.

III. D. 1) Throughput Performance Experiment

Another important metric for consensus algorithms is throughput, which refers to the number of transactions completed by the consensus algorithm within a unit of time. It is generally expressed in TPS and calculated using the following formula.

$$TPS = \frac{T_{\Delta t}}{\Delta t} \tag{14}$$

In this context, Δt represents the time taken to generate a block, and $T\Delta t$ represents the number of transactions completed within the block generation time. A throughput comparison experiment was conducted on five consensus algorithms. Ten nodes were set up for the comparison experiment under the condition of a maximum of 4,000 transactions. Figure 8 shows the experimental results of the throughput performance of the five schemes.

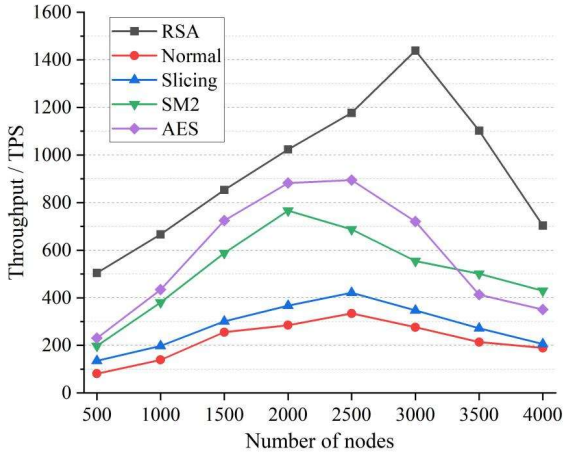

Figure 8: Experimental results of throughput performance for the five schemes

It can be observed that within 2,000 transactions, as transaction volume increases, the consensus throughput gradually increases because the processing capacity of the consensus nodes has not yet exceeded its load limit. After exceeding 3,000 transactions, the consensus throughput begins to decrease, as the processing capacity of the consensus nodes can no longer handle such a large transaction volume. It can be observed that, regardless of whether the transaction volume is within the processing capacity of the consensus nodes or exceeds the threshold, the lack of consideration for the network device capabilities of the nodes during grouping results in weaker processing capacity under high transaction volumes.

The RSA scheme significantly outperforms other schemes at all transaction volumes. At 2,000 transactions, RSA achieves a peak throughput of 1,023.61 TPS, which is 3.6 times that of the Normal scheme (284.64 TPS) and leads SM2 (766.12 TPS) and AES (882.35 TPS). When transaction volume increases to 3,000 transactions, RSA maintains a high performance of 1,438.91 TPS, while other schemes show a significant decline (e.g., Normal drops to 276.16 TPS). For transaction volumes up to 2,000, the throughput of all schemes increases with transaction volume (e.g., RSA increases from 504.38 TPS at 500 transactions to 1,023.61 TPS at 2,000 transactions). Beyond 3,000 transactions, throughput decreases for all schemes due to node overload, but RSA experiences the smallest decline (703.76 TPS at 4,000 transactions), remaining 1.6 times that of SM2 (429.72 TPS) and twice that of AES (350.82 TPS). This result demonstrates that the RSA scheme offers superior scalability and stability under high load conditions.

III. D. 2) Communication Frequency Experiment

The consensus algorithm achieves consensus by enabling nodes to communicate with each other to achieve consistency in steps. The experimental results for the number of communications are shown in Figure 9.

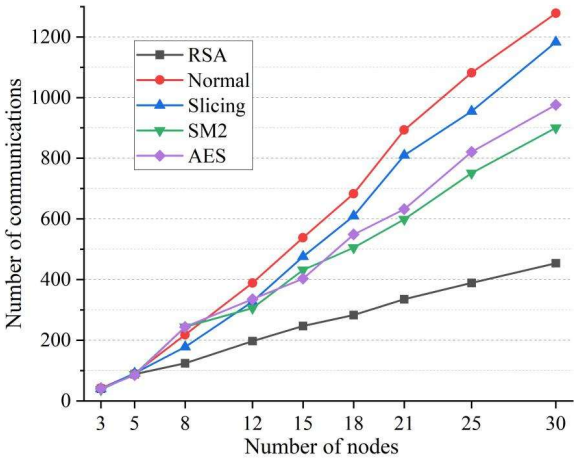

Figure 9: The experimental results of the number of communications

It can be seen that the number of communications in the RSA consensus algorithm is always lower than that of other algorithms. When the number of nodes is 3 and 5, it can be observed that the communication times of the five algorithms are roughly the same. This is because with fewer nodes, consensus efficiency is generally high. Additionally, in this scheme, if there are 5 nodes, only one node can serve as a backup group node, and the advantages of the RSA algorithm cannot be fully demonstrated. As the number of consensus nodes increases, the RSA algorithm requires fewer consensus communications compared to the other four algorithms. At 8 nodes, RSA requires only 124 communications, a 43% reduction from the 218 communications of the Normal algorithm. At 30 nodes, RSA requires only 454 communications, while Normal requires 1,278, SM2 requires 900, and AES requires 976—all significantly higher. RSA reduces the number of communications by 49.6% compared to the next-best SM2. This is because as the number of consensus nodes increases, the consensus synchronization process becomes increasingly complex, and the probability of Byzantine errors occurring at nodes also increases. The RSA algorithm, which has optimized the view switching protocol and uses reliable group nodes to select backup primary nodes to prevent primary node failures, can effectively reduce the number of communications in complex scenarios with a large number of nodes, thereby improving system operational efficiency and reducing consensus overhead. Especially in large-scale node scenarios (≥12 nodes), RSA's communication efficiency advantage becomes evident (e.g., 389 times for RSA vs. 1082 times for Normal at 25 nodes). This result indicates that the RSA scheme can effectively reduce consensus overhead in large-scale node scenarios and improve system efficiency.

III. E. Block recovery success rate

After verifying the throughput and communication efficiency of the consensus algorithm under high loads, it is necessary to further ensure data integrity and recoverability in the event of node failure. Therefore, this section conducts experiments on the block recovery mechanism to evaluate the fault tolerance and recovery success rate of the RSA algorithm-based system under different parameter configurations.

III. E. 1) Impact of compression factor on recovery success rate

On traditional blockchains, a small number of node failures do not affect the integrity of the system, as nodes can download complete ledger data from other nodes via the P2P network. BMC+BIMD also has a certain degree of fault tolerance, enabling data recovery in the event of network node failures. With parameters set to n=25, k=30, and r=1/4, and the replication factor increased from 1 to 5, the block data recovery success rate is shown in Figure 10. When c=1 (i.e., the encoded block set does not use multiple replication methods), it can be observed that as the compression factor increases and the number of generated encoded blocks decreases, the data fault tolerance rate shows a declining trend; However, as the replication factor increases, for example, when the replication factor is c=2/3/4/5, the block recovery success rate reaches 100% in the simulated environment. This is because all network nodes store the encoded block set, and nodes can reconstruct the original block by initiating block recovery requests to other nodes.

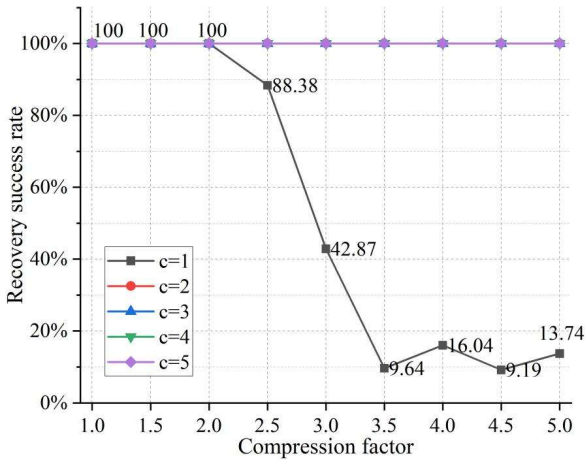

Figure 10: The success rate of restoring block data

Since the primary purpose of introducing the replication factor in this paper is to reduce the number of decoding operations and improve read performance, we evaluate the recovery success rate of directly reconstructing block data under different compression factors. The direct recovery success rate under different compression factors is shown in Figure 11. As can be seen, as the replication factor increases, the probability that the original block shards

are stored across all network nodes also increases. Nodes can download the original block set by sending requests to the target node without undergoing a complex decoding process, thereby increasing the probability of direct block data reconstruction. When the replication factor c = 5, the direct reconstruction recovery success rate under all compression factors is nearly 100%.

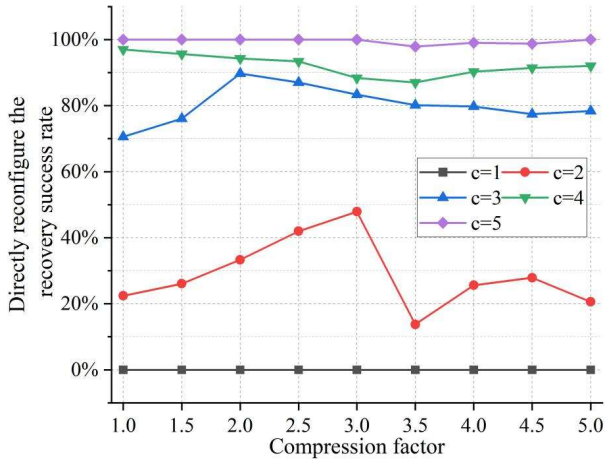

Figure 11: The direct recovery success rate under different compression factors

III. E. 2) Impact of Node Number on Recovery Success Rate

Selecting parameters k=20, l=1.2, and r=1/4, we evaluate the block recovery success rate for different numbers of nodes. The recovery success rate for different numbers of nodes and the direct recovery success rate for different numbers of nodes are shown in Figure 12. The dashed lines represent the recovery success rate for different numbers of nodes, while the solid lines represent the direct recovery success rate for different numbers of nodes. As can be seen, the proportion of fault factors is insufficient to cause the loss of all encoded blocks in the network, resulting in a recovery success rate of 100%. However, the original block set of the block is lost, preventing nodes from directly reconstructing the original block, necessitating recovery through decoding. As the number of nodes increases, the success rate of direct reconstruction recovery decreases. This is because, under a certain fault factor, the number of faulty nodes increases, thereby increasing the probability of loss of the original block set. However, the introduction of the replication factor can improve the probability of direct recovery success. When c=5 and n=40, the direct recovery probability still reaches 89.53%.

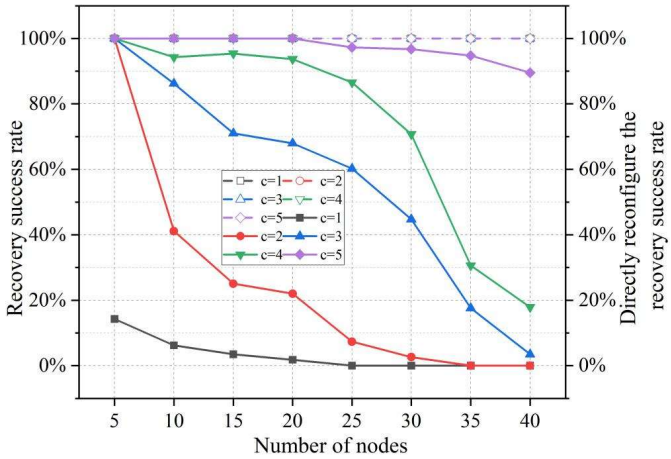

Figure 12: The impact of the number of nodes on the success rate of recovery

IV. Conclusion

This study proposes a cross-institutional educational evaluation data sharing model based on blockchain and RSA accumulators. Systematic experiments using the MOOCCube dataset demonstrate that:

RSA accumulators enable lightweight off-chain verification, with single educational data verification consistently taking 0.285–0.450 seconds, unaffected by increases in attribute count. Even with 50 attributes, verification remains

stable at 0.384 seconds. Key splitting for 200 pairs takes only 62.69 seconds, which is 14.5% faster than the Slicing method.

Bloom filter-supported multi-keyword search in a 5-keyword scenario takes only 4.26 seconds, which is 114.6% faster than the Normal scheme and 67.1% faster than the SM2 scheme, meeting the complex query requirements of educational scenarios. Smart contract management for 50 contracts takes 247.16 seconds, saving 90.57 seconds compared to the standard scheme. Hybrid encryption for 200 courses takes only 99.43 seconds, with efficiency improvements ranging from 7.4% to 12.3%.

The group-optimized consensus mechanism achieves a peak throughput of 1,023.61 TPS with 2,000 transactions, which is 3.6 times that of the standard scheme; The number of communications among 30 nodes is reduced to 454, a decrease of 64.5% compared to the standard solution.

The block recovery mechanism achieves a 100% success rate when the replication factor $c \geq 2$, and the direct recovery rate remains at 89.53% even when the node scale is expanded to 40 nodes, significantly enhancing system robustness.

Acknowledgements

1. Shandong Province Education science "14th Five-Year Plan" project: Research and practice of applying blockchain technology to improve ideological and political education and teaching in secondary vocational schools (Project number: 2021ZC191).

2. Shandong Province 2021 vocational education teaching reform research project: Innovation and practice of teaching mode of ideological and political course in secondary vocational schools based on blockchain (Item number: 2021301).

References

[1] Kustitskaya, T. A., Esin, R. V., Kytmanov, A. A., & Zykova, T. V. (2023). Designing an education database in a higher education institution for the data-driven management of the educational process. Education Sciences, 13(9), 947.

[2] Ang, K. L. M., Ge, F. L., & Seng, K. P. (2020). Big educational data & analytics: Survey, architecture and challenges. IEEE access, 8, 116392-116414.

[3] Kharade, B., & Wagh, K. (2016). Data analytics in educational management system. International Journal of Computer Applications, 975, 8887.

[4] Dietze, S., Sanchez-Alonso, S., Ebner, H., Qing Yu, H., Giordano, D., Marenzi, I., & Pereira Nunes, B. (2013). Interlinking educational resources and the web of data: A survey of challenges and approaches. Program, 47(1), 60-91.

[5] Amo-Filva, D., Fonseca Escudero, D., Sanchez-Sepulveda, M. V., Hasti, H., Aguayo Mauri, S., García-Holgado, A., ... & Paes, C. (2022, January). Open educational resources to enhance students' data protection in schools. In Actas del VII Congreso Internacional sobre Aprendizaje, Innovación y Cooperación (pp. 140-143).

[6] Xu, Y., & Yu, L. (2024, September). Cross-regional Teaching Resource Sharing Solution Based on HADOOP Architecture. In Proceedings of the 2024 International Symposium on Artificial Intelligence for Education (pp. 613-620).

[7] Van Acker, F., Vermeulen, M., Kreijns, K., Lutgerink, J., & Van Buuren, H. (2014). The role of knowledge sharing self-efficacy in sharing Open Educational Resources. Computers in Human Behavior, 39, 136-144.

[8] Yuan, X. (2022). Network education resource information sharing system based on data mining. Mathematical Problems in Engineering, 2022(1), 4080049.

[9] Isus, R., Kolesnikova, K., Khlevna, I., Oleksandr, T., & Liubov, K. (2024). Development of a model of personal data protection in the context of digitalization of the educational sphere using information technology tools. Procedia Computer Science, 231, 347-352.

[10] Alammary, A. S. (2024). Building a Sustainable Digital Infrastructure for Higher Education: A Blockchain-Based Solution for Cross-Institutional Enrollment. Sustainability, 17(1), 194.

[11] Dhara, A. (2023). Building an OER Database: a case study of cross-institutional collaboration. College Libraries, 38(III), 65-73.

[12] Johnston, L. R., Carlson, J., Hudson-Vitale, C., Imker, H., Kozlowski, W., Olendorf, R., ... & Hull, E. (2018). Data curation network: A cross-institutional staffing model for curating research data. International Journal of Digital Curation, 13(1), 125-140.

[13] Zarrin, J., Wen Phang, H., Babu Saheer, L., & Zarrin, B. (2021). Blockchain for decentralization of internet: prospects, trends, and challenges. Cluster Computing, 24(4), 2841-2866.

[14] Ayub Khan, A., Laghari, A. A., Shaikh, A. A., Bourouis, S., Mamlouk, A. M., & Alshazly, H. (2021). Educational blockchain: A secure degree attestation and verification traceability architecture for higher education commission. Applied Sciences, 11(22), 10917.

[15] Wu, X., Meng, T., Huang, H., Liu, L., & Zhang, J. (2024). A Hybrid Consensus Algorithm for Collaborative Protection of Multi-domain Education Data. Journal of Internet Technology, 25(7), 963-975.

[16] Choudhary, A., Chawla, M., & Tiwari, N. (2024). Analyzing functional, technical and bibliometric trends of blockchain applications in education: A systematic review. Multimedia Tools and Applications, 1-46.

[17] Arcinas, M. M. (2021). A blockchain based framework for securing students' educational data. Linguistica Antverpiensia, 2021(2), 4475-4484.

[18] Wang, Y., Sun, Q., & Bie, R. (2022). Blockchain-based secure sharing mechanism of online education data. Procedia Computer Science, 202, 283-288.

[19] Qu, J., & Shao, J. (2024, August). Research on the Construction System of Learning Outcome Certification System Based on Blockchain. In 2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE) (pp. 853-857). IEEE.

[20] Fan, K., Wang, S., Ren, Y., Li, H., & Yang, Y. (2018). Medblock: Efficient and secure medical data sharing via blockchain. Journal of medical systems, 42, 1-11.

[21] Yue, D., Li, R., Zhang, Y., Tian, W., & Huang, Y. (2020). Blockchain-based verification framework for data integrity in edge-cloud storage. Journal of Parallel and Distributed Computing, 146, 1-14.
[22] Li, H., & Han, D. (2019). EduRSS: A blockchain-based educational records secure storage and sharing scheme. IEEE access, 7, 179273-179289.
[23] Tanriverdı, M. (2024). Publiceduchain: A framework for sharing student-owned educational data on public blockchain network. IEEE Access.
[24] Li, Z., & Ma, Z. (2021). A blockchain-based credible and secure education experience data management scheme supporting for searchable encryption. China Communications, 18(6), 172-183.
[25] Abdul Hadi, Z. L., & Au, T. W. (2021). Blockchain for the authentication and immutability of academic credentials issued in Brunei Darussalam. In Computational Intelligence in Information Systems: Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2020) (pp. 75-84). Springer International Publishing.
[26] Reza, A. W., Islam, K., Muntaha, S., Abdur Rahman, O. B., Islam, R., & Arefin, M. S. (2022). Education Certification and Verified Documents Sharing System by Blockchain. International Journal of Intelligent Engineering & Systems, 15(6).
[27] Balobaid, A. S., Alagrash, Y. H., Fadel, A. H., & Hasoon, J. N. (2023). Modeling of blockchain with encryption based secure education record management system. Egyptian Informatics Journal, 24(4), 100411.
[28] Hameed, B., Khan, M. M., Noman, A., Ahmad, M. J., Talib, M. R., Ashfaq, F., ... & Yousaf, M. (2019). A review of Blockchain based educational projects. International Journal of Advanced Computer Science and Applications, 10(10).