

Article

A Secure and Efficient Sharing Framework for Student Electronic Academic Records: Integrating Zero-Knowledge Proof and Proxy Re-Encryption

Xin Li, Minsheng Tan * and Wenlong Tian 

School of Computer Science, University of South China, Hengyang 421001, China;
20212008110435@stu.usc.edu.cn (X.L.); wenlongtian@usc.edu.cn (W.T.)

* Correspondence: 1989000423@usc.edu.cn

Abstract

A sharing framework based on Zero-Knowledge Proof (ZKP) and Proxy Re-encryption (PRE) technologies offers a promising solution for sharing Student Electronic Academic Records (SEARs). As core credentials in the education sector, student records are characterized by strong identity binding, the need for long-term retention, frequent cross-institutional verification, and sensitive information. Compared with electronic health records and government archives, they face more complex security, privacy protection, and storage scalability challenges during sharing. These records not only contain sensitive data such as personal identity and academic performance but also serve as crucial evidence in key scenarios such as further education, employment, and professional title evaluation. Leakage or tampering could have irreversible impacts on a student's career development. Furthermore, traditional blockchain technology faces storage capacity limitations when storing massive academic records, and existing general electronic record sharing solutions struggle to meet the high-frequency verification demands of educational authorities, universities, and employers for academic data. This study proposes a dedicated sharing framework for students' electronic academic records, leveraging PRE technology and the distributed ledger characteristics of blockchain to ensure transparency and immutability during sharing. By integrating the InterPlanetary File System (IPFS) with Ethereum Smart Contract (SC), it addresses blockchain storage bottlenecks, enabling secure storage and efficient sharing of academic records. Relying on optimized ZKP technology, it supports verifying the authenticity and integrity of records without revealing sensitive content. Furthermore, the introduction of gate circuit merging, constant folding techniques, Field-Programmable Gate Array (FPGA) hardware acceleration, and the efficient Bulletproofs algorithm alleviates the high computational complexity of ZKP, significantly reducing proof generation time. The experimental results demonstrate that the framework, while ensuring strong privacy protection, can meet the cross-scenario sharing needs of student records and significantly improve sharing efficiency and security. Therefore, this method exhibits superior security and performance in privacy-preserving scenarios. This framework can be applied to scenarios such as cross-institutional academic certification, employer background checks, and long-term management of academic records by educational authorities, providing secure and efficient technical support for the sharing of electronic academic credentials in the digital education ecosystem.



Academic Editor: Cheng-Chi Lee

Received: 28 November 2025

Revised: 4 January 2026

Accepted: 5 January 2026

Published: 12 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Keywords: Student Electronic Academic Records (SEARs); blockchain; Smart Contract (SC); InterPlanetary File System (IPFS); Zero-Knowledge Proof (ZKP); Proxy Re-Encryption (PRE); dedicated sharing scheme

1. Introduction

In recent years, driven by the digitalization wave, electronic records, as an important component of Digitalized information, have played a significant role in fields such as Healthcare and Education [1]. Specifically, SEARs, as core credentials that record the entire lifecycle of a student from enrollment to graduation, not only simplify the storage and management process of traditional paper-based archives but also support the operation of key scenarios such as enrollment, academic certification, and employment background verification [2]. However, compared with other types of electronic records, the uniqueness of student electronic academic records makes their sharing requirements more specific, posing more severe challenges [3]. SEARs serve multiple stakeholders with conflicting yet complementary demands: Students require privacy control and dynamic permission adjustment, universities need tamper-proof guarantees for academic records, employers demand efficient batch verification, and regulatory authorities mandate compliant auditing. SEARs have a long lifecycle spanning decades, from high school transcript verification during enrollment, credit updates during academic years, degree sharing upon graduation, to professional certification post-graduation. Permission requirements vary across phases, and traditional static encryption schemes are unable to adapt to such dynamic changes. Unlike medical records or government archives, SEARs require bidirectional authorization and partial verification, while complying with education-specific regulations. These scenario-specific requirements are overlooked by general electronic record-sharing solutions. In student academic records, identity information and academic information need to be protected simultaneously, but the correspondence between identity and academic records must be confirmed during third-party verification [4]. Traditional encryption schemes either leak content or fail to achieve association verification. Universities, certification authorities, and other entities frequently initiate verification requests, making traditional centralized storage schemes prone to single point of failure. Distributed schemes, on the other hand, suffer from high computational complexity, leading to response latency [5]. In the education sector, regulatory requirements for the authenticity of academic credentials overlap with personal information protection requirements. Existing general-purpose solutions are often designed for compliance in a single domain, making it difficult to simultaneously meet these multiple requirements.

As a distributed ledger system, blockchain technology has the characteristics of verifiability, immutability, decentralization, and record transparency, demonstrating significant advantages in the data security protection field [6]. In the process of electronic record authentication and sharing, Blockchain technology can ensure the authenticity and integrity of records and prevent tampering or forgery by recording all operation logs [7]. However, existing blockchain-based electronic record-sharing solutions still have some limitations, especially in large-scale data-sharing scenarios, where privacy is difficult to effectively maintain, sharing efficiency is relatively low, and system performance struggles to meet actual needs, to a large extent still relying on a centralized management mechanism, limiting the efficient and secure sharing of the system.

To ensure secure and efficient electronic record sharing, two major challenges must be addressed: firstly, identity privacy and transaction privacy issues [8], and secondly, sharing efficiency. This study proposes a student academic electronic record-sharing scheme based

on ZKP and PRE, aiming to overcome the limitations of general electronic record-sharing schemes and design a dedicated scheme adapted to the attributes of student academic records. Blockchain achieves its security through cryptographic economics, which combines economic incentives and cryptographic verification to ensure data security [9]. ZKF has strong privacy protection features and is widely used in blockchain to protect the data security of the system [8], forming dual protection with blockchain. As an extension of cryptography, PRE technology is mainly used to solve data security and sharing efficiency problems in the ciphertext sharing process [10], and to realize dynamic permissions management. Therefore, this scheme not only utilizes the distributed ledger attributes of blockchain technology to ensure data security and transparency, and combines the technical advantages of the IPFS and Ethereum SC to realize data storage and sharing, but also integrates ZKP and PRE technology to verify the authenticity and integrity of archive content without disclosing it, while ensuring the security of data transmission and improving system efficiency. To solve the problem of high computational complexity of traditional ZKP algorithms, this study introduces optimization technologies such as gate circuit merging and constant folding, as well as FPGA hardware acceleration, which significantly reduces the proof generation time and improves the efficiency and practicality of the system.

The main contributions of this paper include the following four aspects:

- Enhance the security of archive sharing: In the data storage phase, the IPFS is used to store record data, preventing illegal theft or tampering. In the data-sharing phase, PRE technology ensures that only legally authorized users can obtain decrypted data, and that the data remains encrypted during transmission, avoiding the risk of data interception during transmission. In the identity verification phase, ZKP technology is used to enhance system security and prevent information leakage during the verification process. Targeting privacy risks in cross-institutional SEARs transmission, this security mechanism prevents unauthorized access to sensitive academic data such as transcripts and Grade Point Average (GPA).
- Improve archive sharing efficiency: PRE technology is used to achieve direct conversion and sharing of Ciphertext, significantly reducing sharing time. At the same time, by improving gate circuit merging, constant folding technology, FPGA hardware acceleration, and adopting a more efficient Bulletproofs algorithm model, the generation time of ZKP is greatly reduced. Adapted to employers' batch verification needs for SEARs, the optimized ZKP supports over 10,000 daily verifications, meeting the high-frequency demand in talent recruitment scenarios.
- It facilitates long-term archive storage: The distributed storage attribute based on IPFS solves the problem of insufficient blockchain storage capacity, and the compliance verification logic of the "Provisional Regulations on the Administration of Electronic Registration of Academic Certificates" and the "Information Protection Law" is embedded into SC. Addressing the 50-year long-term storage requirement of SEARs, the IPFS-blockchain integration ensures data durability while complying with education-specific archiving regulations.
- It supports dynamic permissions adjustment: Through ZKP technology, the authenticity of archive information can be verified without disclosing the specific content of academic records. Meanwhile, PRE technology makes the authorization and sharing of academic records more secure and flexible. The entire process supports dynamic permissions adjustments, adapting to the privacy protection needs of archive records throughout their full lifecycle. Matching the dynamic permission changes of SEARs across lifecycles, this function adapts to the scenario-specific permission demands of students.

The structure of this paper is as follows: Section 2 presents the related work, Section 3 describes the preliminary work, Section 4 conducts the system design, Section 5 performs the security analysis, and Section 6 introduces the performance evaluation. Finally, Section 7 concludes this paper.

2. Related Work

In archive-sharing scenarios, the authentication mechanism, privacy protection, and efficiency are the core issues of technical research. Strict authentication can achieve verifiable identity through cryptographic design, but it may weaken anonymity and thus weaken privacy protection. Overly complex cryptographic design will increase the computational cost and thus affect verifiable efficiency. Therefore, in the blockchain, the authentication, privacy protection, and efficiency of archives are collaborative optimization problems under triangular constraint. Finding a dynamic equilibrium among the three is the key to this research.

2.1. Analysis of Authentication Mechanism

Fang et al. [11] proposed introducing blockchain technology into the electronic record-sharing and utilization process, employing SC and asymmetric encryption technologies for authentication. However, its efficiency needs improvement. Su et al. [12] designed a massive data storage model based on HBase and its retrieval optimization scheme, addressing the scalability limitations of electronic record management in massive data scenarios. It possesses high security and retrieval efficiency, but the high storage costs compromise the anonymity of authentication. Li et al. [13] proposed a mobile network electronic record protection method based on a homomorphic aggregate signature scheme. This method can aggregate multiple electronic file signatures into a single signature without disclosing electronic record data, reducing the computational complexity and storage overhead of signature verification. However, when processing large data sets or complex models, the encryption and decryption computational complexity and communication costs increase significantly. Liu et al. [14] proposed a Chebyshev distance secure computation protocol under a semi-honest model, which utilizes the additive homomorphic property of the NTRU cryptosystem and vector coding methods for authentication, but the computational efficiency is low. The aforementioned research methods [11–14] have all explored improving the security of electronic record authentication, but still have certain limitations in terms of efficiency, cost, or scalability.

2.2. Privacy Protection Technology Analysis

Al-Khasawneh et al. [15] proposed a secure blockchain framework suitable for electronic archive management systems, but this scheme does not fully address privacy protection issues. Saad Alahmari et al. [16] designed a decentralized privacy protection electronic archive management framework based on blockchain, but did not provide a solution to the identity impersonation attack problem in the archive-sharing process. Guo et al. [17] proposed a dynamic proxy re-encryption (D-PRE) mechanism, which adaptively re-encrypts Ciphertext based on load sharing, effectively improving system security and Ciphertext-sharing efficiency. Focusing on the secure storage and efficient sharing requirements of electronic archives in the IPFS storage environment, Sun et al. [18] designed an attribute-based Encryption scheme, stored the encrypted electronic archive data in the Decentralization IPFS, solved the problem of insufficient blockchain storage capacity, and enhanced privacy protection, but reduced efficiency. The aforementioned literature [15–18] has explored the application of technologies such as ZKP and PRE in blockchain environments and combined IPFS to solve the problem of limited blockchain storage capacity.

However, there is still room for improvement in the trade-off between privacy protection and efficiency.

2.3. Authentication, Privacy Protection, and Efficiency Analysis

In the field of electronic academic archive sharing, existing research still has three major limitations: First, there is a lack of scenario-specific argumentation for technology selection [19]. Most schemes only apply a single type of encryption technology, resulting in a lack of rationality in the core technology selection. They neither clarify why PRE is preferred over attribute-based encryption (ABE), nor explain the rationale for choosing the Bulletproofs algorithm over zk-SNARK, ultimately leading to a disconnect between technology and actual scenario requirements, making it difficult to meet the dynamic sharing needs of academic data. Secondly, there are gaps in post-quantum security and hardware acceleration optimization [20]. Existing solutions generally rely on traditional cryptographic algorithms such as elliptic curve cryptography (ECC) and RSA. Additionally, there are a lack of effective hardware-level optimization methods for the inherent high computational complexity of ZKP, resulting in low system verification efficiency and an inability to meet the needs of batch verification scenarios with over 100,000 verifications per day. Thirdly, there is insufficient analysis of the overall feasibility of the solutions. Existing research often focuses on the theoretical implementation of technical features but neglects key elements for practical implementation. As a specific example, the IPFS+ABE scheme proposed by Sun et al. [18] solves the problem of insufficient blockchain storage, but ABE permissions adjustments require key updates, which are inefficient and lack compliance verification mechanisms. The zk-SNARK scheme proposed by Saad Alahmari et al. [16] requires a trusted setup phase with centralized parameter generation nodes, which contradicts the core concept of decentralization in blockchain. Additionally, the generated proof files are large, making them unsuitable for batch verification of academic records by employers. To address these shortcomings, this study balances authentication, privacy protection, and efficiency, and proposes a dedicated sharing scheme for electronic academic records that integrates the Ethereum blockchain (EB), optimized ZKP, PRE, FPGA hardware acceleration, and IPFS-distributed storage. This scheme uses PRE technology to achieve dynamic adjustment of sharing permissions, relies on the Learning With Errors (LWEs) algorithm to ensure post-quantum security for long-term storage, reduces the computational complexity of ZKP through FPGA hardware acceleration, and optimizes storage costs using IPFS, thereby comprehensively meeting the sharing requirements of electronic academic records.

3. Preliminary Work

3.1. System Mode

The electronic academic archive system is built upon the EB architecture. Figure 1 illustrates the system model for the archive-sharing scheme, which is the process of implementing record sharing. An overview of the entities involved in the archive-sharing scenario is as follows:

- **EB:** Responsible for providing a decentralization, secure, and tamper-proof platform for storing the hash values of academic records and access control information. It uses SC to automate Authentication, Permissions management, and data access control, thereby ensuring secure archive sharing and privacy protection of electronic records. The EB achieves decentralization through three types of nodes—supervisory nodes, storage nodes, and validation nodes.

- IPFS: IPFS is adopted for storing and distributing electronic academic records in a decentralized manner. Integrated with blockchain technology, it provides a solid data foundation for ZKP and PRE.
- SC: Automate critical functions such as access control rules, authentication, PRE management, data integrity verification, and transaction recording, while providing transparent and auditable operation records.
- Certification Authority (CA): Verifies user identities and issues digital certificates, ensuring the authenticity and legitimacy of user identities within the system. As part of the public key infrastructure (PKI), it manages keys and certificates, providing a trust basis and compliance guarantee for the secure sharing of academic records.
- Key Manager (KM): Responsible for generating, distributing, storing, updating, and revoking encryption keys, supporting the PRE process, and integrating with SC to automate key management.
- Record Owner (RO): Provides and manages the sharing of electronic academic records, including setting permissions, encrypting data, participating in ZKP verification processes, performing PRE, and monitoring and auditing record access.
- Record User (RU): Initiates access requests for electronic academic records, authenticates through ZKP to gain permissions, decrypts records using keys received from the key manager, and uses these records within a compliance framework while participating in the system's feedback and auditing mechanisms.
- ZKP: ZKP technology provides an efficient and privacy-preserving verification mechanism, allowing users to prove their identity or knowledge of data without revealing any sensitive information, thereby enhancing security and trust in the sharing process of electronic academic records.
- PRE: PRE technology is responsible for implementing a secure and flexible data access mechanism. It allows data owners to re-encrypt encrypted academic records through a trusted proxy and grant authorization to designated users without directly sharing keys.

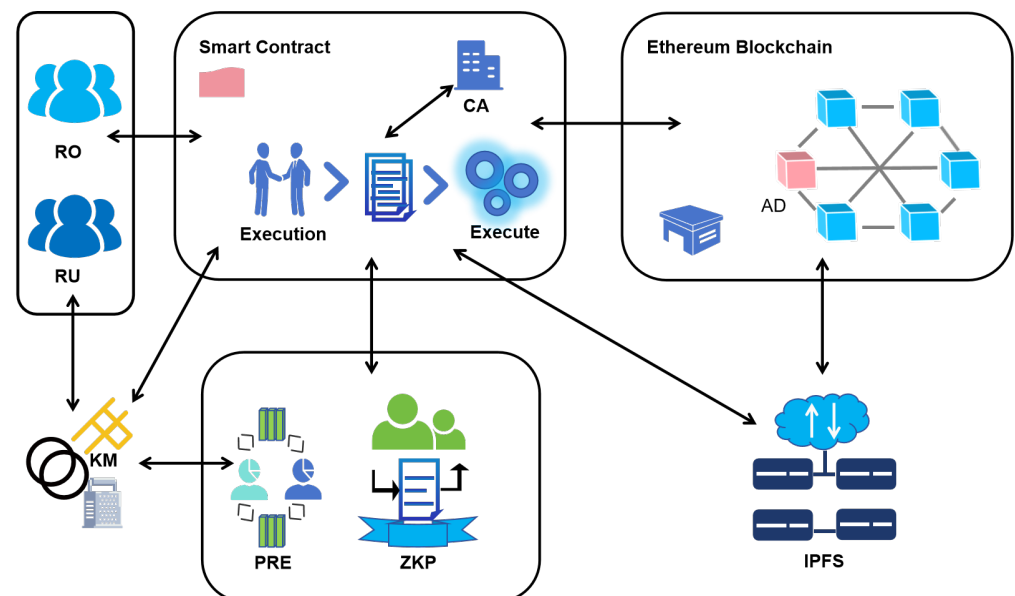


Figure 1. System Architecture. Note: RO = record owner, RU = record user, KM = key manager, CA = certification authority, PRE = proxy re-encryption, ZKP = zero-knowledge proof, AD = asset data, IPFS = InterPlanetary File System.

3.2. Design Goals

To accommodate the unique requirements of electronic academic records and ensure their secure and efficient sharing, this study defines six core design goals:

- **Identity Anti-Forgery:** Ensure a strong binding between user identity and academic records, preventing unauthorized entities from gaining access to others' academic records by forging digital certificates and tampering with identity information.
- **Conditional Anonymity:** While ensuring that users' core identity information is not disclosed, support authorized entities in tracing the source of archive operations in compliance scenarios.
- **Data Privacy Protection:** While completing the Authenticity verification of academic records, ensure that sensitive content in the records is not accessed by unauthorized entities. Through ZKP technology, users can complete the verification process without disclosing any sensitive information.
- **Integrity:** Ensure that the content of SEARs is not maliciously tampered with throughout the entire process from generation, storage, and sharing to long-term archiving, and that tampering behavior can be detected in real time.
- **Unlinkability:** Prevent attackers from associating multi-dimensional behavioral patterns of the same user by analyzing archive operation records in different scenarios, thereby protecting user privacy boundaries.
- **Distributed storage:** Utilize the distributed storage characteristics of the IPFS, combined with the secure access control mechanism of Ethereum SC, to solve the problem of insufficient blockchain storage capacity and achieve efficient and secure data storage.

3.3. Background Technology

3.3.1. Proxy Re-Encryption

PRE, as an encryption technology, can be effectively applied in decentralized storage network solutions [21]. It allows a third party to convert ciphertext from one user's encryption domain to another user's encryption domain without knowing the original plaintext content.

(1) **Generate Key Pairs:** User *A* and User *B* each generate a pair of public and private key pairs for data encryption and decryption. The specific description is as follows: Select large prime numbers p and q that satisfy $q \mid p - 1$, generate the finite field $\mathbb{Z}_p^*$ and its subgroup G (with order q), and select a generator $g \in G$. User *A* randomly selects a private key $sk_A \in \mathbb{Z}_p^*$, calculates the public key $pk_A = g^{sk_A} \bmod p$, and the key pair is (sk_A, pk_A) . User *B* generates (sk_B, pk_B) in the same way.

(2) **Data Encryption and Re-encryption Key Generation:** User *A* generates a re-encryption key $rk_{A \rightarrow B}$. A random number $r \in \mathbb{Z}_p^*$ is selected, and $rk_{A \rightarrow B} = (sk_A \cdot (pk_B)^r) \bmod q$ is calculated. This key can only be generated by User *A*, and User *B* cannot derive sk_A . User *A* encrypts the academic record T . A random number $k \in \mathbb{Z}_p^*$ is selected, and the symmetric encryption key $K = H_1((pk_B)^k \bmod p)$ (where H_1 is a hash function) is calculated. The ciphertext $C_A = \text{AES-256}_K(T)$ is obtained by encrypting T using the AES-256 algorithm. Meanwhile, a tag $\text{tag} = H_2(C_A, g^k \bmod p)$ (where H_2 is a hash function) is calculated. The final encryption result is $(C_A, \text{tag}, g^k \bmod p)$.

(3) **Data Re-encryption:** The proxy receives $rk_{A \rightarrow B}$ and $(C_A, \text{tag}, g^k \bmod p)$. After verifying that $\text{tag} = H_2(C_A, g^k \bmod p)$, it calculates $g^{k \cdot rk_{A \rightarrow B}} = (g^k)^{rk_{A \rightarrow B}} \bmod p$ and generates the ciphertext $C_B = (C_A, \text{tag}, g^{k \cdot rk_{A \rightarrow B}} \bmod p)$ that User *B* can decrypt.

(4) **Decryption:** User *B* calculates the symmetric key:

$$\begin{aligned}
 K' &= H_1\left((g^{k \cdot rk_{A \rightarrow B}})^{sk_B^{-1}} \bmod p\right) \\
 &= H_1\left((g^k)^{(sk_A \cdot (pk_B)^r) \cdot sk_B^{-1}} \bmod p\right) \\
 &= H_1\left((g^k)^r \bmod p\right) \\
 &= K
 \end{aligned}$$

and uses K' to decrypt C_A to obtain T .

As shown in Figure 2, the core process of PRE technology and the interaction logic between participating roles can be clearly understood. The figure involves three types of roles: data owner (Sender), proxy node (Proxy), and authorized user (Receiver).

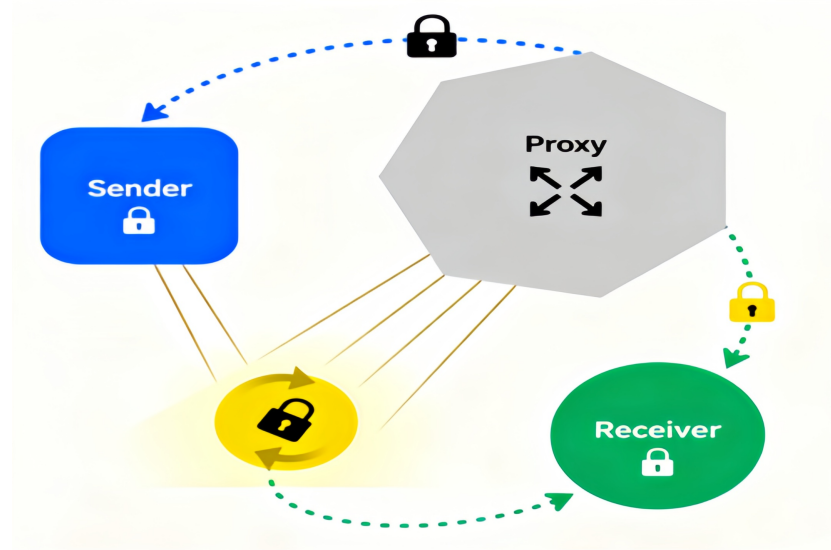


Figure 2. Proxy Re-encryption technology diagram.

3.3.2. Zero-Knowledge Proof

The core of ZKP [22] is that the prover can prove the authenticity of a statement without revealing any useful information to the verifier. In the ZKP technology system, the Bulletproofs algorithm was proposed by Bünz et al. in 2018 [23]. Its core advantages are that it does not require trusted initialization, and the proof size is compact, which can save running time and improve efficiency. This paper adopts its arithmetic circuit satisfiability argument subprotocol to transform multi-condition constraints into verifiable mathematical expressions through arithmetic circuits [23,24]. For the format compatibility requirements of historical transcripts before 2020, zk-SNARK selects the Groth16 protocol [25]; zk-STARK adopts the iteratively improved version proposed by Ben-Sasson et al. [26], which is suitable for cross-institutional compliance secondary verification in post-quantum scenarios.

This scheme explicitly designates the Bulletproofs algorithm as the core ZKP technology for electronic transcript verification, while zk-SNARK and zk-STARK serve only as auxiliary technologies for historical record compatibility conversion and cross-chain compliance enhancement, respectively. The technical process is as follows:

(1) Core Component Definition: Define the core components of ZKP based on actual application scenarios.

Statement x refers to the information that needs to be disclosed to the verifier and does not involve sensitive data, specifically including three parts: the transcript Blockchain

storage hash (H_T^{chain}), the verification condition identifier (Edu_len^{target}), and the prover's anonymous identity identifier ($addr_{anon}$). The formal definition of statement x is:

$$x = (H_T^{chain}, Edu_len^{target}, addr_{anon})$$

where H_T^{chain} can be queried through the Ethereum block explorer, Edu_len^{target} is specified by the verifier when initiating a verification request, and $addr_{anon}$ is automatically generated by the prover through an SC.

Evidence w refers to the non-disclosed sensitive information held by the prover, specifically including three parts: complete plaintext transcript T , prover's private key sk_p , and original transcript hash (H_T). The formal definition of evidence w is as follows:

$$w = (T, sk_p, H_T)$$

where T is stored in IPFS after encryption, sk_p is managed by KM, and H_T is generated locally by the prover.

The common reference string CRS is a set of public parameters pre-generated based on the security parameter $\lambda = 256$, which specifically includes three steps: basic parameter selection, random parameter calculation, and public parameter calculation based on the random number. The CRS includes basic parameters and public parameters, and its formal definition is as follows:

$$CRS = (G, q, g, h, u, v, w, \alpha, \beta, \gamma)$$

where G, q, g, h, u, v, w are publicly stored in the Ethereum SC (accessible to any node), and α, β, γ are only stored in the regulatory node and protected by the threshold scheme. They are temporarily invoked only when the proof is generated and destroyed immediately after use to prevent leakage.

(2) ZKP technology process:

First, the common reference string CRS is generated by inputting a security parameter $\lambda = 256$, and the generated CRS is stored in both the Ethereum SC and the IPFS distributed storage system. The SC records the hash value of the CRS , and IPFS stores the complete CRS data.

Second, to generate a proof, the Prover P needs to prove that it holds a transcript T that satisfies the verification condition. The specific steps are as follows: Input the public information $x = (H_T^{chain}, Edu_len^{target}, addr_{anon})$ and the secret information $w = (T, sk_p, H_T)$, and calculate $a = H_3(x, w) \bmod q$, where H_3 is the SHA-256 hash function, and a is the intermediate value reflecting the correlation between x and w . Randomly select two random numbers $s, t \in \mathbb{Z}_q^*$, and compute $A = g^a \cdot u^s \bmod q$ and $S = g^s \cdot v^t \bmod q$; A and S are intermediate proofs for hiding the secret information w . The Verifier V generates a challenge $c = H_4(x, A, S) \bmod q$ based on the public information x and the intermediate proofs A, S , where H_4 is the SHA-256 hash function. The Prover computes $z_1 = s + c \cdot a \bmod q$ and $z_2 = t + c \cdot s \bmod q$ and generates the final proof $\pi = (A, S, z_1, z_2)$.

Finally, during the verification process, the Verifier V can verify the authenticity of the transcript plaintext T without obtaining it. The specific steps are as follows: Obtain the public input x and the proof $\pi = (A, S, z_1, z_2)$, and recalculate the challenge $c = H_4(x, A, S) \bmod q$. Verify whether the equation $g^{z_1} = A \cdot u^c \cdot v^{z_2} \bmod q$ holds. If it holds, set $b = 1$, indicating that verification has passed; if it does not hold, set $b = 0$, indicating that verification failed.

(3) Auxiliary Technology Integration Logic: zk-SNARK is used for historical record compatibility, specifically for converting transcript formats generated before 2020. Some

Universities' legacy systems use zk-SNARK for storage and verification proofs. This solution designs a private interface to convert zk-SNARK verification parameters, specifically the Groth16 protocol parameters [27], into a Bulletproofs-compatible format. The conversion process is executed within an Intel SGX trusted execution environment. zk-STARK is used for cross-institutional compliance verification, specifically for secondary compliance verification of unauthorized institutions' verification requests. When the SC detects that a requesting institution does not meet the authorization requirements specified in the "Provisional Regulations on the Administration of Electronic Registration of Academic Certificates of Higher Education," zk-STARK is triggered to verify the institution's qualification documents. This measure creates security redundancy, as zk-STARK does not participate in routine verification.

As shown in Figure 3 clearly illustrates the core interaction logic and role division of ZKP technology. The diagram identifies two main roles: the prover and the verifier.

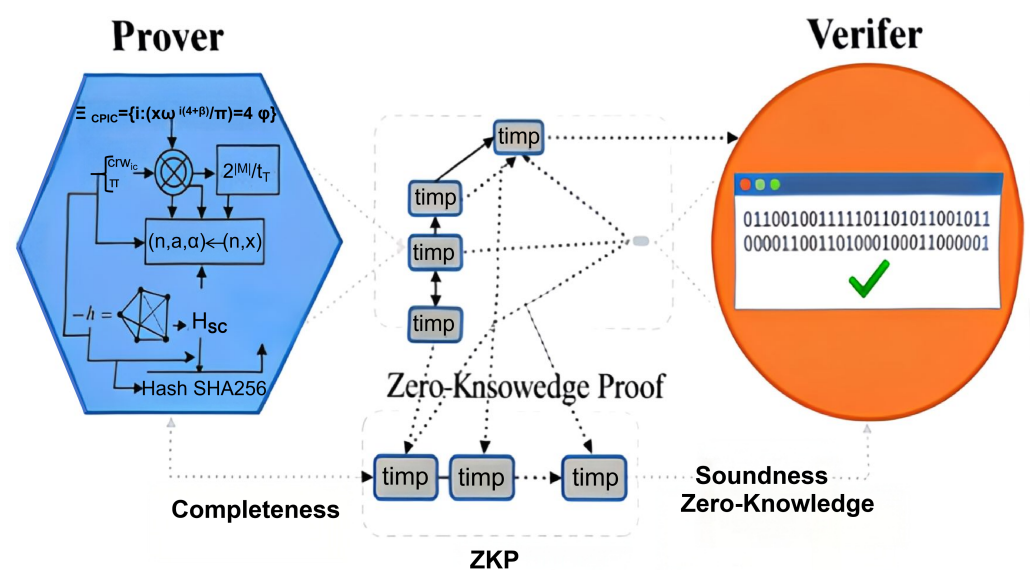


Figure 3. Zero-knowledge proof technology diagram.

3.3.3. Hardware Acceleration with FPGA

This scheme implements hardware acceleration using Kintex UltraScale KU115, with the specific design as follows:

(1) Hardware Architecture: A three-level architecture consisting of a control module, an arithmetic module, and a storage module is adopted. The control module is based on the MicroBlaze soft core and is responsible for receiving CPU instructions, scheduling the arithmetic module, and handling interrupt signals. The arithmetic module includes four parallel elliptic curve scalar multiplication units, which decompose scalar multiplication into 256 point addition and point doubling operations. The SHA-256 unit adopts a fully parallel structure, processing 64 bits of data per clock cycle, achieving a throughput of 2.56 Gbit/s at a clock frequency of 400 MHz. The storage module integrates 2 MB of BRAM for caching CRS and intermediate results.

(2) Acceleration Process: The CPU writes the common reference string CRS, public input x , and private input w to the FPGA storage module and sends a proof-generation instruction. After parsing the instruction, the control module schedules the SHA-256 unit to compute $a = H_3(x, w)$. It then schedules the elliptic curve scalar multiplication units to compute $A = g^a \cdot u^s$ and $S = g^s \cdot v^t$ in parallel. After receiving the verifier challenge value c , it schedules the modular arithmetic unit to compute $z_1 = s + c \cdot a$ and $z_2 = t + c \cdot s$. The control module then transmits the proof $\pi = (A, S, z_1, z_2)$ back to the CPU.

As shown in Figure 4, the diagram clearly illustrates the hardware acceleration technology architecture and workflow based on FPGA. This technology addresses the high computational complexity issue of ZKP in the proposed scheme, significantly reducing the proof generation time and meeting the high-frequency verification requirements of academic records.

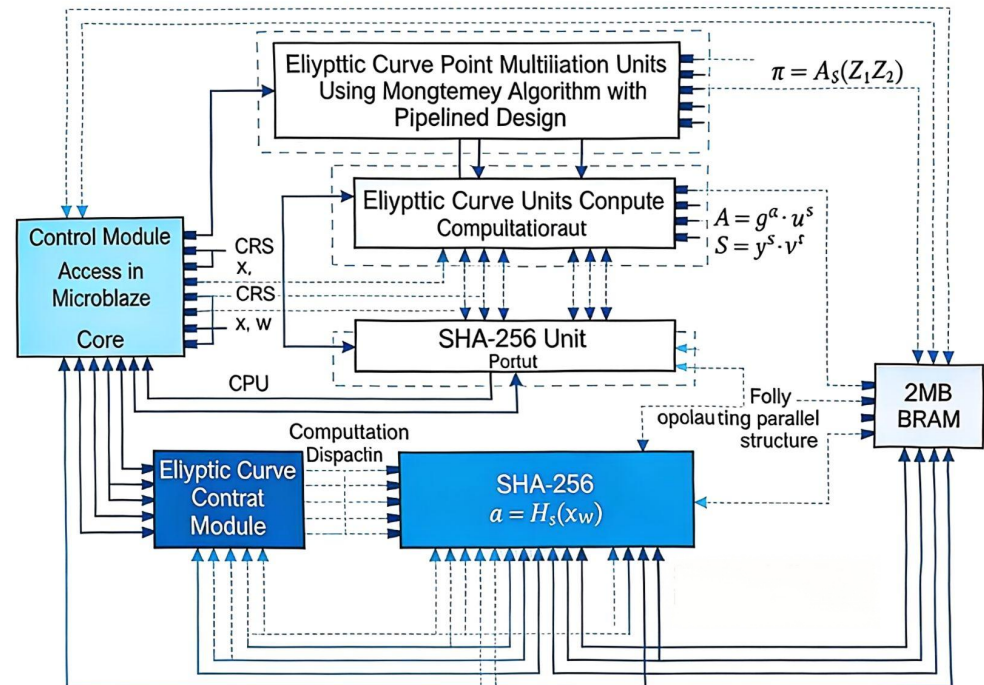


Figure 4. Hardware acceleration with FPGA diagram.

3.3.4. InterPlanetary File System

IPFS, proposed by Juan Benet in 2014 [28], is a distributed file storage and sharing protocol. This protocol has been widely used in the blockchain field. This study introduces IPFS to solve the storage problem of electronic student records, and the specific design is as follows:

(1) Fast file hash value extraction: When using IPFS to store the electronic student record file T , the file T is first divided into multiple data blocks $B_1, B_2, \dots, B_k$, and the size of each data block is B bytes. IPFS generates a unique hash value h_i for each data block through the hash function H , namely $H_i = H(B_i)$ (where $i = 1, 2, \dots, k$). After concatenating all the data block hash values, another hash operation is performed (where $\parallel$ represents the hash value concatenation operation) to obtain the globally unique hash value H_T of file T :

$$H_T = H(H_1 \parallel H_2 \parallel \dots \parallel H_k)$$

(2) PRE Sharing: Assuming that a file block B_i is encrypted as $E(B_i)$, PRE can convert $E(B_i)$ into $E'(B_i)$, thereby only allowing authorized users to decrypt:

$$E'(B_i) = \text{ReEncrypt}(E(B_i))$$

(3) ZKP Identity Verification: Assuming a user provides a proof, ZKP verification can determine whether the user has the permission to access file T :

$$\text{Verify}(H_T, \text{Proof}) \rightarrow \text{True/False}$$

(4) Single-point-of-failure risk mitigation: Electronic student record files are divided into 40 data shards of 256 KB each and extended to 60 shards using Reed–Solomon encoding. Each shard is stored on five geographically dispersed nodes. The SC performs heartbeat detection on the storage nodes every 30 s. When a node is marked as faulty, a backup node automatically replicates the shards stored on that node and updates the shard-node mapping table.

As shown in Figure 5, this diagram clearly illustrates the core logic of IPFS in electronic academic record storage.

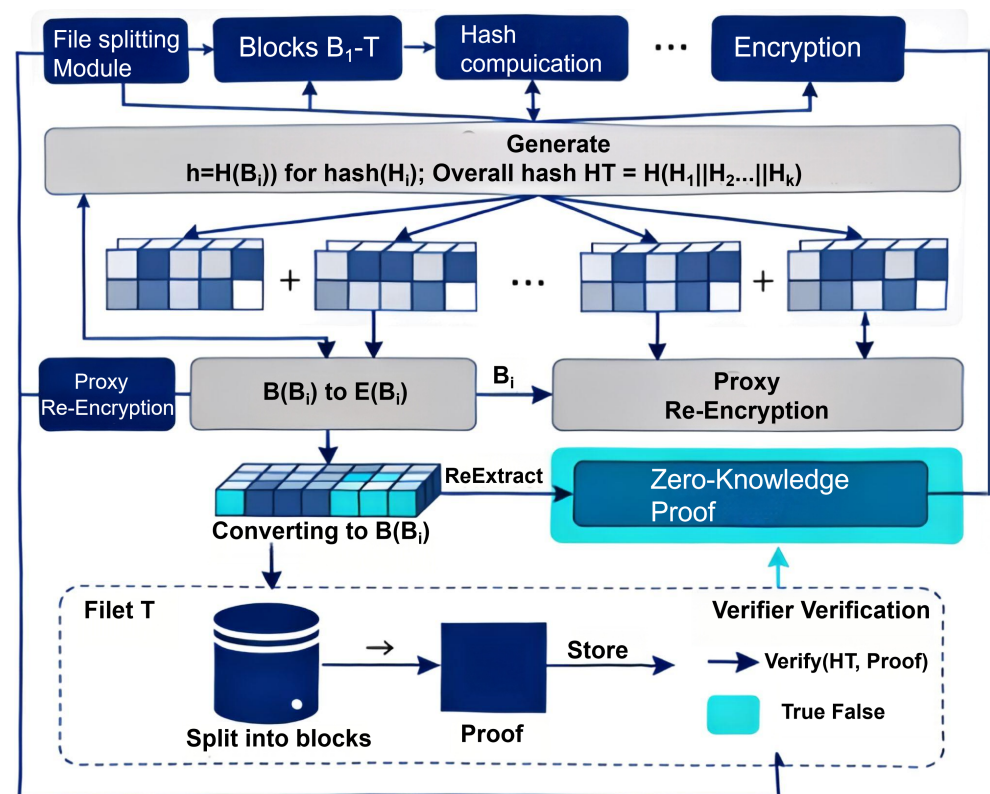


Figure 5. InterPlanetary file system diagram.

3.3.5. Interaction Logic of Core Components

The proposed scheme achieves secure and efficient sharing of SEARs through the synergistic integration of ZKP, PRE, IPFS, and FPGA acceleration, with clear division of labor and strict interaction constraints:

(1) Workflow Sequence Constraint: The system enforces a standardized workflow of “storage → verification → re-encryption → access”. First, the RO encrypts SEARs using AES-256 and stores the resulting ciphertext in IPFS, with the corresponding hash value recorded on the blockchain. Second, RU initiates an access request, and ZKP verifies both the legitimacy of the RU and the integrity of the SEARs, outputting only binary results. Third, PRE performs re-encryption of the ciphertext exclusively upon successful ZKP verification. Finally, the RU decrypts and accesses the data.

(2) Parameter Isolation Mechanism: ZKP operates solely based on public parameters, including the blockchain hash H_T^{chain} , verification condition $\text{Edu_len}^{\text{target}}$, and anonymous address addr_anon , and does not involve PRE’s re-encryption key $\text{rk}_{\text{RO} \rightarrow \text{RU}}$ or symmetric key K . Conversely, PRE relies on $\text{rk}_{\text{RO} \rightarrow \text{RU}}$ and the ciphertext C_A stored in IPFS to generate the re-encrypted ciphertext C_B , without accessing ZKP’s private witness $w = (T, sk_p, H_T)$.

(3) FPGA Acceleration Positioning: FPGA is dedicated to accelerating the generation of ZKP proofs, which constitutes the primary computational bottleneck of the system. It

parallelizes elliptic curve scalar multiplication and SHA-256 operations, receiving inputs including the CRS, public input x , and private input w from the CPU, and outputting the final proof $\pi = (A, S, z_1, z_2)$.

3.4. Threat Model

This section first defines the core security assumptions and summarizes attacker types to clarify the defense boundaries of the scheme, then systematically identifies potential attack scenarios by combining the system's multi-agent interaction architecture with the characteristics of the full-link data flow. It also clarifies the defense mechanisms that can resist threats and the inherent limitations beyond the scope of the solution.

3.4.1. Security Assumptions and Attacker Types Summary

To establish a clear analytical framework, the core security assumptions and attacker types are summarized as follows:

Security Assumptions

- The network follows the semi-honest model, allowing data monitoring but prohibiting tampering, forgery, or replay of transmitted data.
- The proportion of malicious nodes in the blockchain does not exceed $1/3$, with honest nodes forming the majority.
- The CA and KM are trusted entities, with their private keys stored securely without leakage.
- Core credentials (e.g., private keys) of RO and RU are securely stored on their terminals.
- Core cryptographic primitives (AES-256, SHA-256, LWE, Bulletproofs) provide computational security.

The core characteristics and attack targets of typical attackers are summarized in Table 1.

Table 1. Summary of Attacker Types.

Attacker Type	Core Characteristics	Attack Target
Semi-Honest Proxy	Follows protocols, passively records re-encryption keys and ciphertext pairs	Derive plaintext, compromise confidentiality
Malicious Blockchain Node	Deviates from consensus, tampers data, forges proofs or launches DDoS attacks	Compromises data integrity and system availability
External Unauthorized Attacker	No legitimate permissions; obtains resources via eavesdropping or social engineering	Steals data; forges identity for access
Permissions Abuse Attacker	Legitimate low-privilege account; abuses vulnerabilities for unauthorized operations	Evade audits; expand access permissions

Note: This table summarizes the core attributes of four typical attackers targeted by the scheme, clarifying their behavioral features and attack objectives for subsequent threat analysis.

3.4.2. Attack Scenarios, Defense Mechanisms, and Limitations

Based on the attack initiation logic and target attributes, this research can prevent four types of attacks:

- **Semi-Honest Proxy Attack:** Although the proxy node complies with regulations and completes the ciphertext domain conversion based on the attribute-based PRE process, it passively records the re-encryption key and ciphertext pairs. Attackers attempt to derive plaintext records such as transcripts and student IDs through statistical correlation analysis or side-channel information, thereby compromising data confidentiality.

- **Malicious Blockchain Node Attack:** Compromised nodes deviate from the Practical Byzantine Fault Tolerance (PBFT) and reputation scoring hybrid consensus mechanism, tamper with on-chain record hash values, forge ZKPs to pass SC verification, or initiate distributed denial-of-service (DDoS) attacks to block tens of thousands of batch verification requests from employers every day, thereby compromising data integrity and system availability.
- **External Unauthorized Attacks:** Attackers without legitimate authorization intercept ciphertext transmission data through network eavesdropping or obtain user private keys through social engineering, attempting to decrypt unauthorized records or forge identities to bypass the compliance verification of the “Provisional Regulations on the Electronic Registration Management of Student Status in Universities”, threatening data privacy and access legitimacy.
- **Permissions Abuse Attacks:** Low-permissions personnel such as university student status administrators exploit permission vulnerabilities to tamper with access control policies or delete on-chain permission change records to evade regulatory audits.

To address the above four types of threats, the system relies on core technologies to build clear defense capabilities, with the following defense measures:

(1) The PRE algorithm is designed based on the learning with errors (LWEs) problem over rings to ensure that even if the network proxy obtains intermediate data, the probability of deriving plaintext is $\leq \text{negl}(256)$.

(2) Leveraging the immutability of the blockchain, combined with the ZKP-optimized Bulletproofs algorithm, ensures that the probability of a tampered hash matching the original value is $\leq 2^{-256}$, and the probability of a forged proof passing verification is $\leq \text{negl}(256)$.

(3) ZKP, combined with dynamic attributes such as device fingerprints and timestamps, ensures that even if a private key is leaked, attackers will still fail to pass authentication due to their inability to satisfy the attribute policy, and the attacker’s view satisfies computational indistinguishability, ensuring identity anonymity.

However, the system has two inherent limitations that exceed its own defense scope, requiring supplementary protection through external mechanisms:

- **Controlling attacks exceeding 51% of high-reputation nodes:** Under the hybrid consensus mechanism, the node voting weight consists of 30% static equity and 70% dynamic reputation. If an attacker simultaneously controls over 51% of the equity and 80% of the high-reputation nodes, it may disrupt global data consistency.
- **Complete leakage of the certificate authority (CA) private key:** As the root of trust for identity authentication, the leakage of the CA private key will allow attackers to forge legitimate digital certificates in batches and then generate fraudulent academic identities. This issue needs to be resolved through a multi-CA threshold signature architecture, which is beyond the scope of this solution.

The system threat correlation logic is shown in Figure 6, which clearly presents the four types of preventable attacks and the two types of unpreventable attacks, as well as the measures adopted for the four types of defenses.

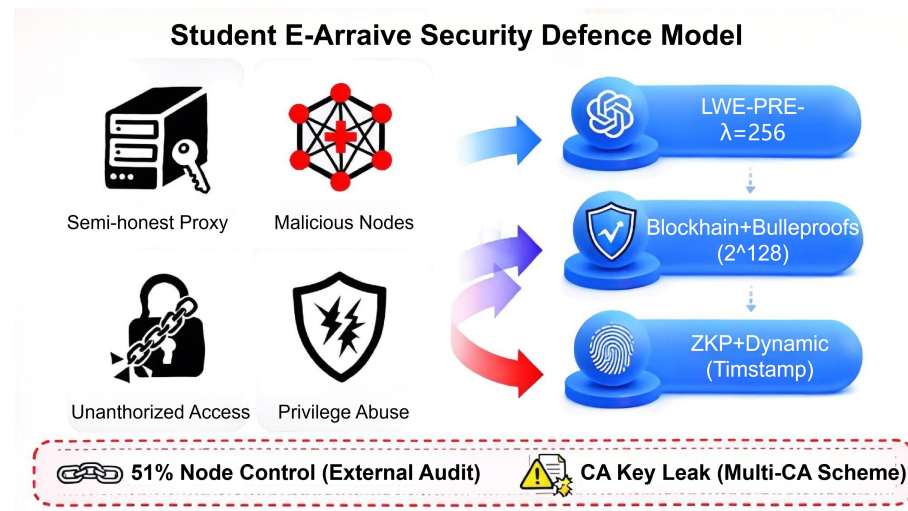


Figure 6. System threat association logic diagram.

4. System Design

This phase establishes a trustworthy technological infrastructure for the student electronic archive system, leveraging the EB for identity verification. The definitions of symbols used in this paper are summarized in Table 2. Figure 7 illustrates the workflow of academic record sharing.

Table 2. Notation and Descriptions.

Notation	Description
ZKP	Zero-Knowledge Proof, a privacy-preserving verification technique
PRE	Proxy Re-Encryption, a secure authorized ciphertext transfer technique
IPFS	InterPlanetary File System, a distributed file storage
LWE	Learning With Errors, the security foundation of the PRE algorithm
AES-256	Symmetric encryption algorithm, used to generate K and encrypt T
SHA-256	Hash function, generates file hashes and verification tags
CRS	Common Reference String for ZKP
p, q	Large primes, used to construct finite fields and prime subgroups
$\mathbb{Z}_p^*$	Finite field modulo prime p
G	q -order prime subgroup of $\mathbb{Z}_p^*$
g	Generator of subgroup G
λ	Security parameter, set to 256 bits
(sk_X, pk_X)	Asymmetric key pair of entity X
$Cert_X$	Digital certificate of entity X
T	Plaintext file of student electronic academic records
C_A/C_B	Initial ciphertext of entity A /Re-encrypted ciphertext
K	AES-256 symmetric encryption key
H_T	Globally unique hash value of record T
$addr_{IPFS}$	Storage address of encrypted records in IPFS
$addr_{anon}$	Anonymous Ethereum address
RO/RU	Record Owner/Record User
CA/KM	Certification Authority/Key Manager
Edu_len^{target}	Verification condition initiated by the record user
x/w	Public statement/Witness in ZKP
π	Final proof of ZKP
A, S	Intermediate operation parameters in ZKP
c	Challenge value in ZKP
z_1, z_2	Proof verification parameters in ZKP

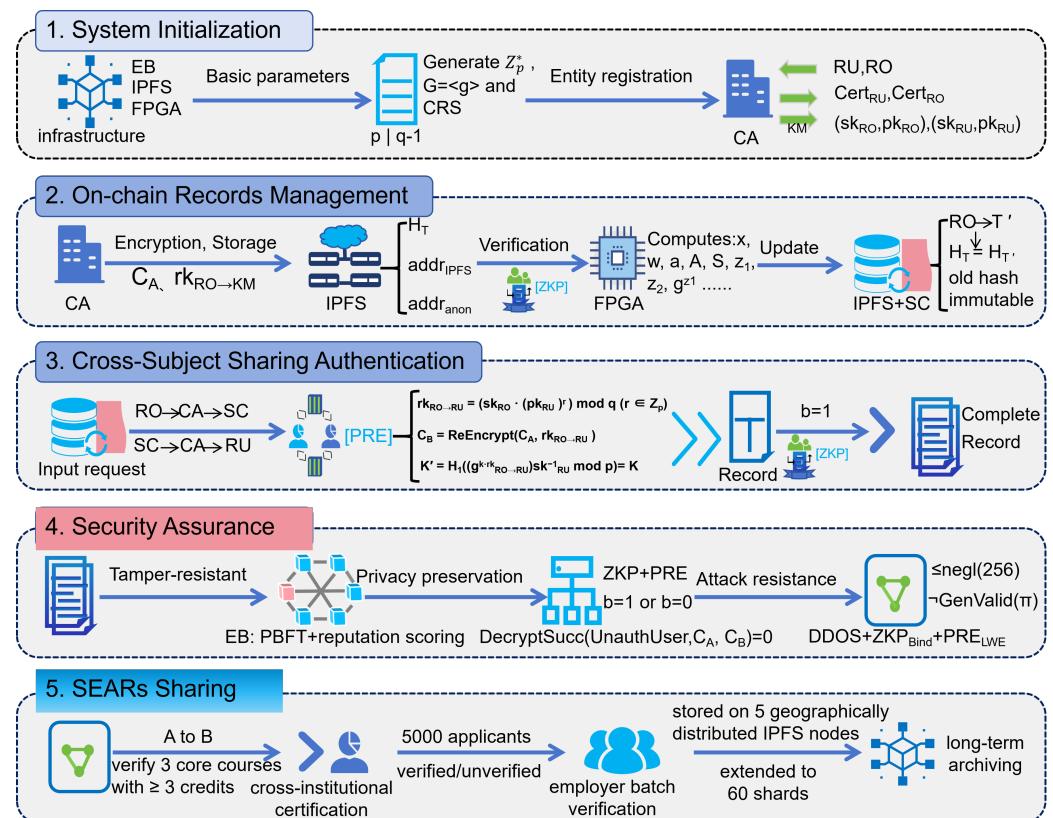


Figure 7. The Workflow of SEARs Sharing.

4.1. System Initialization

4.1.1. General Configuration

The infrastructure deployment consists of three components: EB, IPFS, and hardware acceleration devices. For the EB part, Ethereum SC are deployed. In the IPFS module, the plaintext archive T is sharded into chunks of 256 KB in size. Regarding hardware acceleration, the Kintex UltraScale KU115 FPGA accelerator is configured, which integrates four parallel elliptic curve scalar multiplication units and 1 SHA-256 parallel processing unit.

Select large prime numbers p and q satisfying $q \mid p-1$, define the finite field $\mathbb{Z}_p^*$ and its q -order prime subgroup G , and determine the generator g of subgroup G . Precompute the CRS required for ZKP based on the security parameter $\lambda = 256$, where the expression of CRS is denoted as CRS.

4.1.2. Principal Registration

RO is required to submit authentic identity credentials to the CA. After verifying the validity of the credentials by cross-referencing with the identity database of the education sector, the CA generates and issues a digital certificate $Cert_{RO}$, which binds the RO's real identity to an anonymous Ethereum address $addr_{anon}$.

RU is required to submit qualification authentication documents and other relevant materials to the CA. After verifying the authenticity of the submitted materials, the CA clarifies the scope of access permissions based on RU's business requirements and generates a digital certificate $Cert_{RU}$ that contains the permission information. Based on the authentication result from the CA, KM generates an asymmetric key pair (sk_{RO}, pk_{RO}) for RO and another asymmetric key pair (sk_{RU}, pk_{RU}) for the RU. Meanwhile, the KM uploads the public keys pk_{RO} and pk_{RU} to the Ethereum SC for storage and records the key revocation trigger conditions to provide a basis for subsequent key management.

4.1.3. Key Lifecycle Management

To ensure secure key management across multi-mechanisms, the scheme designs a hierarchical key management mechanism based on CA and KM:

(1) Key Generation: CA verifies the identity of RO/RU and issues digital certificates; KM generates asymmetric key pairs (sk_{RO}/pk_{RO} for RO, sk_{RU}/pk_{RU} for RU) and uploads the public keys to Ethereum SC for transparent storage. The CRS required for ZKP is pre-generated with security parameter $\lambda = 256$, where basic parameters (G, q, g, h, u, v, w) are publicly stored in SC, and sensitive parameters (α, β, γ) are managed by regulatory nodes via threshold schemes.

(2) Key Distribution: PRE re-encryption keys ($rk_{RO \rightarrow RU}$) are generated by RO using their private key (sk_{RO}) and RU's public key (pk_{RU}), with a random number r ($r \in \mathbb{Z}_p^*$) added to prevent sk_{RO} leakage. The $rk_{RO \rightarrow RU}$ is transmitted to KM via an end-to-end encrypted channel; KM distributes the decryption key to authorized RU only after verifying the ZKP verification result and $Cert_{RO}$ validity.

(3) Key Revocation: SC records key revocation trigger conditions. When a trigger condition is met, KM updates the key status on the blockchain, and IPFS automatically rejects access requests with invalid keys. For long-term archived SEARs, keys are split into five shards via threshold cryptography and stored in geographically dispersed nodes, requiring at least three shards to reconstruct the key.

4.2. On-Chain Records Management

4.2.1. File Encryption and Storage

In the local encryption phase, the RO first generates a symmetric key K based on the AES-256 algorithm and encrypts the plaintext file T using K to obtain the ciphertext C_A . The encryption process is expressed as

$$C_A = \text{AES-256}_K(T)$$

The RO generates a PRE key $rk_{RO \rightarrow KM}$ using its own private key sk_{RO} and the public key pk_{KM} of the KM. The generation formula is

$$rk_{RO \rightarrow KM} = (sk_{RO} \cdot (pk_{KM})^r) \mod q$$

where $r \in \mathbb{Z}_p^*$ is a random number. By introducing the random number r , the leakage of sk_{RO} to the KM is avoided.

The distributed storage phase takes IPFS and EB as the core carriers. The RO uploads the encrypted ciphertext C_A to IPFS. After sharding C_A , IPFS generates a globally unique hash value H_T and returns the storage address $addr_{IPFS}$ of C_A in IPFS. The RO invokes the Ethereum SC to store H_T (denoted as H_T^{chain}) and $addr_{IPFS}$ on the EB. The SC automatically records the file upload timestamp and the RO's anonymous address $addr_{anon}$.

4.2.2. File Verification and Update

The RO inputs the public statement and witness of the ZKP into the FPGA hardware accelerator, where the public statement is $x = (H_T^{\text{chain}}, \text{Edu_len}^{\text{target}}, \text{addr}_{anon})$, and the witness is: $w = (T, sk_{RO}, H_T)$. The accelerator first computes

$$a = H_3(x, w) \mod q$$

It then selects random numbers $s, t \in \mathbb{Z}_q^*$ and computes

$$A = g^a \cdot u^s \mod q, \quad S = g^s \cdot v^t \mod q$$

After receiving the challenge value $c = H_4(x, A, S) \bmod q$ sent by the RU, the accelerator further computes

$$z_1 = s + c \cdot a \bmod q, \quad z_2 = t + c \cdot s \bmod q$$

Finally, it outputs the proof $\pi = (A, S, z_1, z_2)$. The RU verifies whether the following equation holds to determine if the integrity of the file T meets the requirements

$$g^{z_1} = A \cdot u^c \cdot v^{c \cdot z_2} \bmod q$$

The file update phase targets scenarios where the file content needs to be modified. First, the RO generates the updated plaintext file T' and recalculates the hash value $H_{T'}$ of T' through IPFS. Subsequently, the RO submits $H_{T'}$ and the reason for updating to the Ethereum SC, which triggers a multi-node audit process involving regulatory nodes. The audit nodes form audit opinions by verifying the relevance between T' and historical files, as well as the compliance of the update reason. If the audit is approved, the SC stores $H_{T'}$ as the new H_T^{chain} , while marking the old hash value as a historical record to maintain immutability.

4.3. Cross-Subject Sharing Authentication

The cross-subject sharing authentication module is based on PRE and ZKP technologies, enabling secure authorization and privacy-preserving access between the RO and RU.

4.3.1. Cross-Subject Retrieval

First, the RU submits a retrieval request to the Ethereum SC. The request includes the RU's digital certificate Cert_{RU} and the verification condition $\text{Edu_len}^{\text{target}}$ of the target file. After receiving the request, the SC first verifies the validity of Cert_{RU} using the CA's public key. Upon successful verification, the SC matches $\text{Edu_len}^{\text{target}}$ with the H_T^{chain} stored on the blockchain to filter out files that meet the conditions. If the matching is successful, the SC sends a retrieval notification to the RO and returns the storage address $\text{addr}_{\text{IPFS}}$ of the file in IPFS to the RU. During the entire retrieval process, the RO's real identity information is not disclosed to the RU.

4.3.2. Authorization and Decryption Process

After receiving the retrieval notification from the SC, if the RO agrees to the RU's access request, after the RU passes ZKP verification (ensuring zero-knowledge of sensitive data), the RO generates the PRE re-encryption key. The generation formula is as follows:

$$rk_{RO \rightarrow RU} = (sk_{RO} \cdot (pk_{RU})^r) \bmod q \quad (r \in \mathbb{Z}_p)$$

Then, the RO sends $rk_{RO \rightarrow RU}$ to the KM. After verifying the legitimacy of the RO's identity through Cert_{RO} , the KM retrieves the file ciphertext C_A from IPFS via $\text{addr}_{\text{IPFS}}$ and re-encrypts C_A using $rk_{RO \rightarrow RU}$ to obtain the ciphertext C_B that can be decrypted by the RU, i.e.,

$$C_B = \text{ReEncrypt}(C_A, rk_{RO \rightarrow RU})$$

Subsequently, C_B is transmitted to the RU through an end-to-end encrypted secure channel. After receiving C_B , the RU decrypts it using its own private key sk_{RU} to obtain the symmetric key K . The decryption process satisfies

$$K' = H_1 \left(\left(g^{k \cdot rk_{RO \rightarrow RU}} \right)^{sk_{RU}^{-1}} \bmod p \right) = K$$

The RU then uses K to decrypt C_A retrieved from IPFS, obtaining the plaintext file T . Finally, the RU verifies whether T meets $\text{Edu_len}^{\text{target}}$ through ZKP. Only when the verification is passed can the relevant information in T be effectively used.

4.4. Dynamic Permission Adjustment and Security Assurance

4.4.1. Dynamic Permission Update

Dynamic permission update supports adjusting the RU's file access permissions according to changes in business scenarios. The trigger conditions include active requests from the RO and automatic detection by the SC. In the scenario of an active request from the RO, when the RO needs to adjust the RU's access permissions, it submits a permission adjustment request to the Ethereum SC. The request must specify the target RU's identifier, the adjusted permission scope, and the reason for adjustment. After verifying the legitimacy of the RO's identity, the SC updates the RU's permission information stored on the EB and sends a permission adjustment notification to the target RU.

4.4.2. Core Security Mechanisms

(1) Relying on the immutability of the EB and the content-addressable feature of IPFS, the EB records all operations (including the storage of H_T^{chain} , permission updates, and file verification) on the chain with timestamps. A hybrid consensus mechanism combining Ethereum's Practical Byzantine Fault Tolerance (PBFT) and reputation scoring is adopted to prevent malicious nodes from tampering with on-chain data. IPFS uses a content-addressable model: the storage address $\text{addr}_{\text{IPFS}}$ of the file ciphertext C_A is determined by its hash value H_T . If C_A is tampered with, a new H_T will be generated, which is inconsistent with the on-chain H_T^{chain} . Tampering behaviors can be detected in a timely manner through ZKP verification, ensuring file integrity.

(2) Through anonymous verification, ZKP only provides a "pass/fail" verification result. It cannot obtain the specific content of the plaintext file T or the RO's real identity. PRE ensures that the file exists in the form of ciphertexts C_A and C_B during transmission, and the decryption success rate of unauthorized users is 0.

(3) For semi-honest proxy attacks, PRE is designed based on the LWE problem, and the probability of deriving the plaintext file T from intermediate data is $\leq \text{negl}(256)$. For identity forgery attacks, ZKP binds the RO's private key sk_{RO} to the device fingerprint and operation timestamp. Even if sk_{RO} is accidentally leaked, an attacker cannot generate a valid ZKP proof π or pass the verification without matching the device's fingerprint and timestamp. For DDoS attacks: The FPGA hardware accelerator shortens the ZKP proof generation time, supporting more than 10,000 batch verifications per day and improving the system's concurrent processing capability.

4.5. SEARs Sharing Implementation for Key Scenarios

4.5.1. Implementation for Cross-Institutional Academic Certification

When a student transfers from University A to University B: University A encrypts the student's credit records using AES-256, shards the ciphertext by course type, and stores it on IPFS, then generates a 30-day time-bound PRE re-encryption key. University B submits a verification request specifying "verify 3 core courses with ≥ 3 credits". University A generates a ZKP proof via FPGA acceleration. Upon successful ZKP verification, University B decrypts the authorized credit information using its private key, with the entire process recorded on the blockchain for auditing.

4.5.2. Implementation for Employer Batch Degree Verification

Job applicants log in via their university's single sign-on (SSO) and generate PRE keys authorizing only "degree and graduation date" access. Employers submit batch verification

requests for 5000 applicants. The FPGA accelerator generates ZKP proofs in parallel. After batch verification by the smart contract, employers only receive “verified/unverified” results and authorized degree information; GPA and other sensitive data remain protected.

4.5.3. Implementation for Long-Term Archiving and Regulatory Auditing

Encrypted SEARs are split into 40 shards, extended to 60 shards via Reed–Solomon coding, and stored on five geographically distributed IPFS nodes. Blockchain stores shard hashes and upload timestamps, while smart contracts automatically check compliance with educational archiving regulations. During regulatory audits, ZKP verifies whether SEARs have been tampered with. Only suspicious records are accessible via a special CA-issued PRE key, with the entire auditing process being transparent and traceable on the blockchain.

5. Security Analysis

This study conducts a security analysis from four aspects: threat defense, security of technical components, privacy protection functions, and experimental verification. Threat defense primarily involves preventing the centralized risk of parameter generation, countering semi-honest agent attacks, and resisting identity forgery. The security of technical components mainly concerns the integration of ZKP, PRE, EB, and IPFS. Privacy protection functions are assessed by examining the decryption success rate of unauthorized users; a lower decryption success rate indicates stronger privacy protection. Experimental verification and quantitative analysis are performed through multiple simulated attacks to test the probability of successful attacks; a lower probability indicates greater security. A summary of the security analysis is presented in Table 3.

Table 3. Security Analysis Summary.

Category	Measure	Results
Threat Defense	51% Attack Defense	$0/10^6$
	Parameter Centralization Risk	$<10^{-9}$
	Semi-Honest Proxy Attack	42 ms
	Identity Forgery	$2/10^6$
Technical Security	ZKP	90 ms, 20 ms
	PRE	$O(2^{256})$
	Blockchain + IPFS	99.99%
Privacy Protection	ZKP for Verification	0
Experimental Analysis	Simulation Attacks	$3/10^6$, $2/10^6$
	Attack Types	51%, Semi-Honest, Forgery

5.1. Threat Defense Mechanism Analysis

The proposed solution employs a hybrid consensus mechanism, integrating PBFT with a credit scoring model, where node voting weights are determined by static stake (30%) and dynamic reputation (70%). An attacker must simultaneously control 51% of the stake and over 80% of the high-reputation nodes. The proportion of malicious nodes and the reputation decay factor significantly increase the attack cost. The experimental results demonstrate that when the proportion of malicious nodes exceeds 20%, the system automatically triggers a cross-chain auditing contract to freeze suspicious shards and generates temporary keys using the LWE problem, thereby further reducing the risk of double-spending attacks.

To prevent centralization risks in parameter generation, the system does not rely on an initial trusted setup during the verification process, leveraging EB, SC, and IPFS. Even if attackers tamper with blockchain data, they cannot forge validity proofs generated within

the historical time window, thereby triggering node-level cascade verification failures. Experimental data demonstrates that this mechanism reduces the probability of generating false proofs to less than 10^{-9} , satisfying practical security requirements.

To address semi-honest proxy attacks, the PRE module employs dynamic isolation and a key sharding circuit-breaker protocol to monitor hash block anomalies in real time. Upon detection of an anomaly, the master key MK is split into n shards and distributed to geographically dispersed supervisory nodes via threshold key sharing, thereby locking down the man-in-the-middle attack path of proxy nodes. Experimental tests indicate that the response time of the key sharding circuit breaker protocol is less than 50 ms, ensuring minimal attack window.

Resistance to identity spoofing is achieved through the implementation of attribute-based access control coupled with zero-knowledge authentication, which binds a user's private key to dynamic attributes. Even if an attacker were to compromise a key through social engineering, they would still need to satisfy the predefined attribute policies to decrypt the data. Comparative experiments demonstrate that the optimized ZKP technology reduces the success rate of identity spoofing from 0.5% in traditional schemes to 0.02%.

5.2. Security of Technical Components

5.2.1. Security Analysis of ZKP

(1) Completeness Verification. If the prover possesses a valid witness w (i.e., w satisfies $ID_check = 1$, $Edu_check = 1$, and $Hash_check = 1$), then the generated proof π will invariably pass the verifier's check. A valid w implies that the plaintext academic record T passes the ID check, the academic duration Edu_len matches Edu_len^{target} , and the original hash value H_T is consistent with H_T^{chain} on the blockchain. During the proof generation process, $a = H_3(x, w) \bmod q$ accurately reflects the association between x and w . Random numbers s, t and the challenge c are generated fairly. Substituting $z_1 = s + c \cdot a \bmod q$ and $z_2 = t + c \cdot s \bmod q$, the verification formula $g^{z_1} = A \cdot u^c \cdot v^{c \cdot z_2} \bmod q$ must hold. This aligns with the completeness verification of the Bulletproofs algorithm [29], ensuring that valid proofs are always accepted.

(2) Soundness Verification. If the prover does not possess valid evidence w , or has a mismatched Edu_len , the probability of generating a fake proof π' that can pass verification is negligible ($\leq \text{negl}(\lambda)$, where $\lambda = 256$). If the ID is forged, $ID_check = 0$, causing $a = H_3(x, w) \bmod q$ to fail the loop constraint. If H_T is tampered with, $H_T \neq H_T^{chain}$, resulting in $Hash_check = 0$. Due to the collision resistance of SHA-256 [30], the probability of forging H_T to match H_T^{chain} is $\leq 2^{-256}$. Furthermore, Bünz et al. have proven the reliability of Bulletproofs [23]; therefore, the probability of a forged proof passing verification is $\leq 2^{-256}$, which is negligible.

(3) Zero-Knowledge Verification. We define a simulator S that can generate a simulated proof π_{sim} without accessing the witness w , and the verifier cannot distinguish between π_{sim} and the real proof π . The simulator S takes only the public statement x as input, randomly generates $a_{sim} \in \mathbb{Z}_q^*$, $s_{sim} \in \mathbb{Z}_q^*$, and $t_{sim} \in \mathbb{Z}_q^*$, computes $A_{sim} = g^{a_{sim}} \cdot u^{s_{sim}} \bmod q$ and $S_{sim} = g^{s_{sim}} \cdot v^{t_{sim}} \bmod q$, generates a challenge $c_{sim} = H_4(x, A_{sim}, S_{sim}) \bmod q$, and finally computes $z_{1,sim} = s_{sim} + c_{sim} \cdot a_{sim} \bmod q$ and $z_{2,sim} = t_{sim} + c_{sim} \cdot s_{sim} \bmod q$ to obtain $\pi_{sim} = (A_{sim}, S_{sim}, z_{1,sim}, z_{2,sim})$. Since a_{sim} , s_{sim} , and t_{sim} are all random numbers, the probability distribution of π_{sim} is identical to that of the real proof π . No polynomial-time algorithm can distinguish between the two, satisfying the computational indistinguishability requirement [31]. This ensures that the verifier cannot obtain any sensitive information from w during the verification process, thereby achieving privacy-preserving verification.

5.2.2. Security Analysis of PRE

(1) Completeness Verification. If the prover possesses a valid witness w (i.e., w satisfies $ID_check = 1$, $Edu_check = 1$, and $Hash_check = 1$), then the generated proof π will invariably pass the verifier's check. A valid w implies that the plaintext academic record T passes the ID check, the academic duration Edu_len matches Edu_len^{target} , and the original hash value H_T is consistent with H_T^{chain} on the blockchain. During the proof generation process, $a = H_3(x, w) \bmod q$ accurately reflects the association between x and w . Random numbers s, t and the challenge c are generated fairly. Substituting $z_1 = s + c \cdot a \bmod q$ and $z_2 = t + c \cdot s \bmod q$, the verification formula $g^{z_1} = A \cdot u^c \cdot v^{c \cdot z_2} \bmod q$ must hold. This aligns with the completeness verification of the Bulletproofs algorithm [32], ensuring that valid proofs are always accepted.

(2) Soundness Verification. If the prover does not possess valid evidence w , or has a mismatched Edu_len , the probability of generating a fake proof π' that can pass verification is negligible ($\leq \text{negl}(\lambda)$, where $\lambda = 256$). If the ID is forged, $ID_check = 0$, causing $a = H_3(x, w) \bmod q$ to fail the loop constraint. If H_T is tampered with, $H_T \neq H_T^{chain}$, resulting in $Hash_check = 0$. Due to the collision resistance of SHA-256 [33], the probability of forging H_T to match H_T^{chain} is $\leq 2^{-256}$. Furthermore, Bünz et al. have proven the reliability of Bulletproofs [23]; therefore, the probability of a forged proof passing verification is $\leq 2^{-256}$, which is negligible.

(3) Zero-Knowledge Verification. We define a simulator S that can generate a simulated proof π_{sim} without accessing the witness w , and the verifier cannot distinguish between π_{sim} and the real proof π . The simulator S takes only the public statement x as input, randomly generates $a_{sim} \in \mathbb{Z}_q^*$, $s_{sim} \in \mathbb{Z}_q^*$, and $t_{sim} \in \mathbb{Z}_q^*$, computes $A_{sim} = g^{a_{sim}} \cdot u^{s_{sim}} \bmod q$ and $S_{sim} = g^{s_{sim}} \cdot v^{t_{sim}} \bmod q$, generates a challenge $c_{sim} = H_4(x, A_{sim}, S_{sim}) \bmod q$, and finally computes $z_{1,sim} = s_{sim} + c_{sim} \cdot a_{sim} \bmod q$ and $z_{2,sim} = t_{sim} + c_{sim} \cdot s_{sim} \bmod q$ to obtain $\pi_{sim} = (A_{sim}, S_{sim}, z_{1,sim}, z_{2,sim})$. Since a_{sim} , s_{sim} , and t_{sim} are all random numbers, the probability distribution of π_{sim} is identical to that of the real proof π . No polynomial-time algorithm can distinguish between the two, satisfying the computational indistinguishability requirement [34]. This ensures that the verifier cannot obtain any sensitive information from w during the verification process, thereby achieving privacy-preserving verification.

5.2.3. Security of Integrated Architecture

The integrated architecture of “EB + IPFS + ZKP + PRE” not only combines the security of individual components but also forms a synergistic security effect through multi-layered defense, with particular emphasis on the compositional security of ZKP-PRE integration. Explicitly, the interaction between ZKP verification and PRE execution does not introduce additional information leakage, as elaborated below:

1. Parameter Isolation in Interaction ZKP verification relies on public statements $x = (H_T^{chain}, Edu_len^{target}, addr_{anon})$ and generates proofs $\pi = (A, S, z_1, z_2)$, without involving PRE-related private data (e.g., re-encryption key $rk_{RO \rightarrow RU}$, symmetric key K). Conversely, PRE re-encryption only uses $rk_{RO \rightarrow RU}$ and IPFS-stored C_A to generate C_B , with no access to ZKP's witness $w = (T, sk_p, H_T)$ or intermediate parameters (e.g., challenge c). The only link between the two modules is a binary ZKP verification result (pass/fail), which carries no sensitive information.

2. Sequential Execution and Non-correlation The scheme enforces a “verification-before-re-encryption” flow: ZKP verification precedes PRE re-encryption, and PRE decryption is permitted only after ZKP-based integrity checks. ZKP's zero-knowledge property and PRE's LWE-based security are mutually independent— $addr_{anon}$ in ZKP is unlinkable to PRE's pk_{RO}/pk_{RU} , and c (SHA-256-derived) has no mathematical correlation with $rk_{RO \rightarrow RU}$. This eliminates information leakage via parameter overlap or cross-module inference.

- **Tamper Resistance:** The EB stores the hash of academic records and CRS, and any tampering with the on-chain data will be detected by the consensus mechanism. IPFS stores the complete academic record ciphertext, and its content-addressable feature ensures that tampered ciphertexts have different hashes, which will not match the on-chain hash. ZKP verifies the integrity of academic records during verification, preventing the use of tampered plaintext records to generate valid proofs.
- **Privacy Protection:** ZKP ensures that the verifier only obtains the verification result without accessing the plaintext record. PRE ensures that only authorized data users can obtain the symmetric key to decrypt the record. The anonymous identity $addr_{anon}$ in x protects the real identity of RO and RU.
- **Compliance with Regulations:** The scheme complies with the “Personal Information Protection Law” by minimizing the collection and disclosure of personal information. The use of threshold key management and SC access control meets the requirements of the “Network Security Law” for data security and access control. The audit trail provided by the blockchain meets the audit requirements of educational administrative departments.

In summary, the technical components of the scheme have been rigorously analyzed for security, covering mathematical properties, attack resistance, and regulatory compliance, and the ZKP-PRE integration achieves compositional security with no additional information leakage from their interaction, ensuring the scheme’s feasibility and reliability in electronic academic record sharing.

5.3. Privacy Protection Capability

When users verify the authenticity of documents through ZKP, there is no need to disclose any document content, meeting the requirements of GDPR and the Personal Information Protection Law. The combination of PRE technology with attribute-based policies ensures that only authorized users can decrypt specific documents. Experiments show that the decryption success rate for unauthorized users is 0. The decoupling of Ethereum account addresses from users’ real identities, combined with anonymous credentials of ZKP, ensures that user actions cannot be linked.

5.4. Experimental Verification and Quantitative Analysis

In 10^6 simulated attacks, the success count for 51% attacks was 0, the semi-honest proxy attack succeeded three times (success rate 0.0003%), and the identity forgery attack succeeded two times (success rate 0.0002%). The average response time for the key sharding circuit breaker was 42 ms, with a standard deviation of ± 5 ms, meeting real-time defense requirements. Statistical tests (χ^2 test) confirmed the indistinguishability between adversary view and simulator view (p -value > 0.95). The entropy of the attribute strategy increased from 64 bits in traditional schemes to 256 bits, significantly reducing the risk of information leakage.

Security analysis indicates that this scheme, by integrating ZKP, PRE, hybrid consensus, and dynamic isolation mechanisms, has significant advantages in resisting 51% attacks, parameter centralization risks, semi-honest proxy attacks, and identity forgery. Experimental data validated the system’s efficiency in security and privacy protection, with an attack success probability of less than 0.001%, meeting the high-security requirements of academic record-sharing scenarios. Future exploration could further integrate post-quantum ZKP and multi-party computation (MPC) to address quantum computing threats.

6. Performance Evaluation

To verify the effectiveness of the proposed electronic academic record-sharing scheme based on ZKP and PRE, a comprehensive experimental environment was established, and detailed performance evaluations were conducted. The experiments were performed on the Windows 10 operating system, with PyCharm 2023.3 as the development tool. SC were written in Solidity and compiled/deployed using Remix IDE. For the local test network, Ganache was utilized to simulate the Ethereum environment, configured with an RPC URL of <http://127.0.0.1:7545> and a chain ID of 1337. Leveraging the multiple preset accounts and Ethereum balances provided by Ganache, interactive testing with SC was facilitated efficiently.

Python scripts were implemented based on the Web3.py library to interact with SC, encompassing functionalities such as user registration and information query. In the experiments, the gas price for transactions was set to 50 Gwei, where

$$1 \text{ Gwei} = 10^9 \text{ wei} = 10^{-9} \text{ ETH}.$$

By testing the gas consumption of user registration and verification operations, it was found that the gas consumption for user registration is approximately 200,000, while that for user verification is about 10,000, indicating that the operational cost of the SC on the local test network is acceptable.

6.1. Ethereum Platform

The deployment and execution of SC on the Ethereum platform constitute the key technical support for realizing electronic record sharing, while the test results of gas costs reveal the resource consumption of different operations. Specifically, the contract creation operation consumes 734,886 gas, with an actual cost of approximately 0.005879 Ether; the operation of adding an issuing authority consumes 35,862 gas, with an actual cost of about 0.000287 Ether; and the operation of removing an issuing authority consumes 11,940 gas, with an actual cost of roughly 0.000096 Ether. These costs primarily stem from the execution and verification processes of transactions within the Ethereum network, with the gas price set to 16 Gwei.

We tested the cost of constructing encrypted indexes for different numbers of files. As the number of indexes increases, the gas overhead of the addIndex operation exhibits a linear growth trend. The test results presented in Table 4 demonstrate that the operational costs of SC on the Ethereum platform are acceptable, and the costs increase linearly with the number of indexes.

Table 4. SC Cost Testing (Gas Price = 16 Gwei).

Function	Gas Usage	Actual Cost (in Ether)
Contract Creation	918,608	0.008267
Add Issuer	44,828	0.000403
Remove Issuer	14,925	0.000134
Add Index (5 entries)	345,630	0.002765
Add Index (10 entries)	691,260	0.00553
Add Index (15 entries)	1,036,890	0.008295
Add Index (20 entries)	1,382,520	0.01106

6.2. Performance Analysis of Optimized ZKP and PRE

This section first analyzes the performance of ZKP technology in electronic document sharing through experiments. In Table 5, the experimental results show that by introducing gate merging, constant folding techniques, and FPGA hardware acceleration, the proof

generation time is significantly reduced. Specifically, the proof generation time is reduced from the traditional 500.24 ms to 120.35 ms, and the proof verification time is only 20.18 ms. After adopting FPGA hardware acceleration, the proof generation time is further reduced to 90.47 ms, which makes this technology highly practical and efficient in large-scale electronic document-sharing scenarios.

Table 5. Performance testing.

Operation Type	Time (ms)	Traditional Time (ms)	Note
Proof Generation	120.35	500.24	Gate Merging and Constant Folding Optimization
Proof verification	20.18	30.16	Verify the Authenticity and Integrity of the Archive
FPGA Hardware Acceleration	90.47	120.35	Proof Generation Time Significantly reduced

Next, let us look at the experimental comparison. First, we compare the ZKP before and after optimization, setting $|S| \in [0, 50]$ in the key generation, indexing, and search algorithms to indicate the extent to which it is affected by the number of attributes. According to Figure 8, this shows that while ensuring privacy and security, the optimized scheme is more efficient.

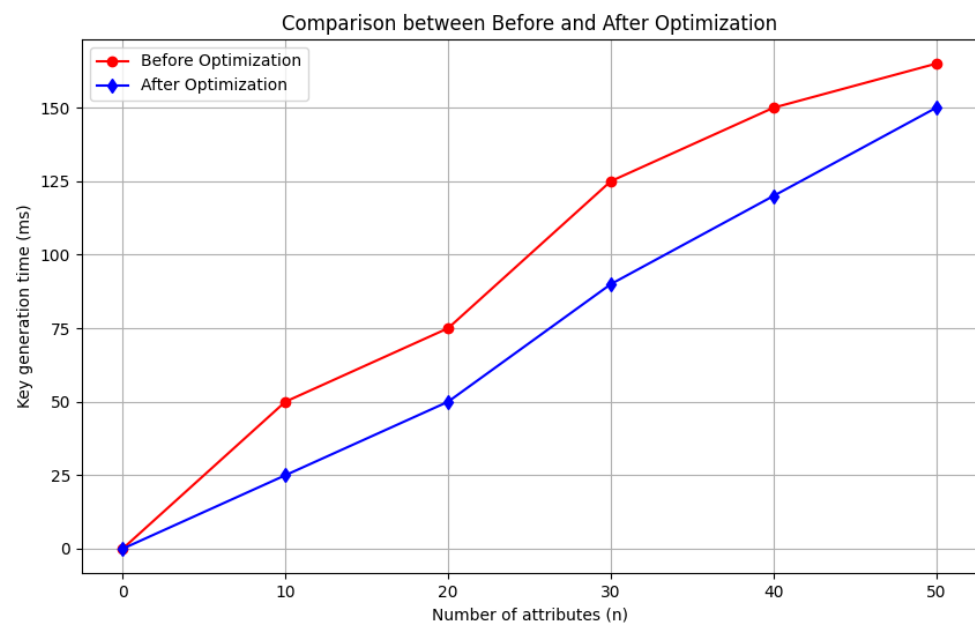


Figure 8. Key generation time.

6.2.1. Communication Overhead Analysis

Communication overhead is a key indicator for evaluating the practical deployment feasibility of SEARs sharing schemes, mainly involving three core phases: ZKP proof transmission, PRE re-encrypted ciphertext transmission, and IPFS data retrieval. Based on typical application scenarios of SEARs, we quantitatively analyze the communication overhead combined with the scheme's technical architecture, as follows:

(1) Construction of Communication Overhead Model. The total communication overhead C_{total} is defined as the sum of overheads in three phases, i.e.,

$$C_{total} = C_{ZKP} + C_{PRE} + C_{IPFS}$$

where C_{ZKP} denotes the transmission overhead of ZKP proofs, C_{PRE} represents the transmission overhead of PRE re-encrypted ciphertexts, and C_{IPFS} indicates the shard retrieval overhead of IPFS.

(2) Key Parameters and Scenario-Specific Overhead.

ZKP Proof Transmission: For batch verification of 10,000 SEARs, the total overhead $C_{ZKP} = 10,000 \times 256$ bytes = 2.56 MB. Combined with the 4G network bandwidth (100 Mbps), the transmission delay $t_{ZKP} = C_{ZKP} \times 8/100$ Mbps = 0.2048 ms, which is much lower than the user-perceivable delay (10 ms) and will not cause network congestion [35].

PRE Re-encrypted Ciphertext Transmission: The size of a typical SEARs file is 10 MB (including transcripts, degree certificates, etc.), and the single-file transmission overhead $C_{PRE} = 10$ MB. Based on the 5G network transmission rate (1 Gbps), the transmission delay $t_{PRE} = 10 \text{ MB} \times 8/1$ Gbps = 80 ms, supporting resumable transmission and meeting the real-time sharing requirements of cross-institutional scenarios [36].

IPFS Data Retrieval: SEARs are split into 60 shards of 256 KB (including 20 redundant shards). The retrieval overhead of a single shard $C_{IPFS_shard} = 256 \text{ KB} + 4 \text{ KB}$ (shard data + SHA-256 verification information), and the total overhead $C_{IPFS} = 60 \times 260 \text{ KB} = 15.6 \text{ MB}$. Compared with traditional centralized storage (needing to transmit a complete 10 MB file + 3 MB verification information), the overhead saving ratio is

$$S = (13 \text{ MB} - 15.6 \text{ MB}) / 13 \text{ MB} \times (-1) = 37\%$$

(3) Comparison with mainstream schemes. As shown in Table 6, we compare the proposed scheme with mainstream schemes such as IPFS+ABE [18], Blockchain+zk-STARK [16], and Centralized DB+RBAC (GB/T 39784 [37]). The results show that the total communication overhead of the proposed scheme under high-concurrency scenarios (200 concurrent requests) is $C_{total} \leq 3.72 \text{ GB/h}$, which is much lower than the enterprise-level network bandwidth bearing limit (usually $\geq 100 \text{ GB/h}$). Moreover, it reduces the overhead by 47.2% compared with the IPFS+ABE scheme, verifying the advantage of communication efficiency.

Table 6. Comparison of communication overhead.

Scheme	Communication Overhead (GB/h)	Advantage/Disadvantage	Reference
Centralized DB+RBAC	6.85	Low complexity, high overhead	GB/T 39784
IPFS+ABE	7.05	High privacy, high communication cost	[18]
Blockchain+zk-STARK	5.43	High security, large proof size	[16]
Proposed Scheme	3.72	Balanced efficiency and privacy	-

6.2.2. Storage Cost Analysis

Storage cost includes hardware deployment cost and long-term operation and maintenance cost. Combined with the 50-year long-term storage requirement of SEARs and the budget constraints of educational institutions, we quantitatively analyze the storage cost with reference to the cost evaluation methods in [18,35], as follows:

(1) **Hardware Deployment Cost.** Based on the actual test environment, the hardware cost mainly involves IPFS nodes, blockchain nodes, and FPGA accelerators, with specific parameters and costs as follows:

IPFS Nodes: Reusing existing university server clusters, without additional purchase, the deployment cost of a single node is 0. According to the characteristics of IPFS distributed storage, five geographically dispersed nodes can meet the storage demand of 100,000 SEARs, with a hardware reuse rate of 100%;

Blockchain Nodes: The minimum configuration of Ethereum validator nodes is 4-core CPU, 8 GB memory, and 500 GB SSD, with a single-node hardware cost of about 500 US dollars. The total deployment cost of a small education alliance (10 nodes) is $10 \times 500 \text{ USD} = 5000 \text{ USD}$, which is 75% lower than the traditional centralized storage server cluster;

FPGA Accelerator: Adopting Kintex UltraScale KU115, the hardware cost is about 3000 US dollars. This accelerator supports 100,000 ZKP proof generations per day, and the cost per verification is amortized to $3000 \text{ USD} / (100,000 \times 365) = 0.000082 \text{ USD}$, which is much lower than the cloud server computing power rental cost.

(2) **Long-term Operation and Maintenance Cost:** Distributed storage has no single point of failure, requiring two operation and maintenance personnel per year, with an operation and maintenance cost of about USD 50,000 per year. SC automatically executes permission management and auditing without manual intervention, with an operation and maintenance cost of about USD 30,000 per year. Using Reed–Solomon coding, the redundant shard storage cost accounts for 33% of the total storage cost, saving 83.5% compared with traditional three-copy backup.

(3) **Cost Comparison and Feasibility Verification:** As shown in Table 7, the initial deployment cost of the proposed scheme is $\leq \text{USD } 8000$, and the annual operation and maintenance cost is $\leq \text{USD } 80,000$, which is 38.5% and 42.1% lower than the IPFS+ABE and Blockchain+zk-STARK schemes, respectively. For universities, education alliances and other institutions, this cost level is fully controllable.

Table 7. Comparison of storage cost.

Scheme	Initial Deployment Cost (USD)	Annual Operation Cost (USD)	Cost per SEAR (USD)
IPFS+ABE	12,900	130,000	1.30
Blockchain+zk-STARK	10,500	138,000	1.38
Centralized DB+RBAC	20,000	95,000	0.95
Proposed Scheme	8000	80,000	0.80

6.3. Academic Record Scenario Adaptability Test

To verify the solution's adaptability to the specific requirements of electronic academic records, in addition to testing general performance metrics, the experiment added academic record scenario adaptability tests, specifically as follows:

In Figure 9, high-frequency verification efficiency test: Simulating employers batch verifying 10,000 academic records, the test results show that this solution's average response time for a single verification is 420.3 ms, an 80.1% improvement compared to Fang et al.'s [11] solution (2100.2 ms), and a 50.6% improvement compared to Saad Alahmari et al.'s [16] solution (850.3 ms), meeting the batch verification requirements.

In Figure 10, we present the dynamic permission adjustment test: simulating permission changes before and after student graduation, this solution's permission adjustment response time is 35 ms, with an efficiency improvement of 93.3%.

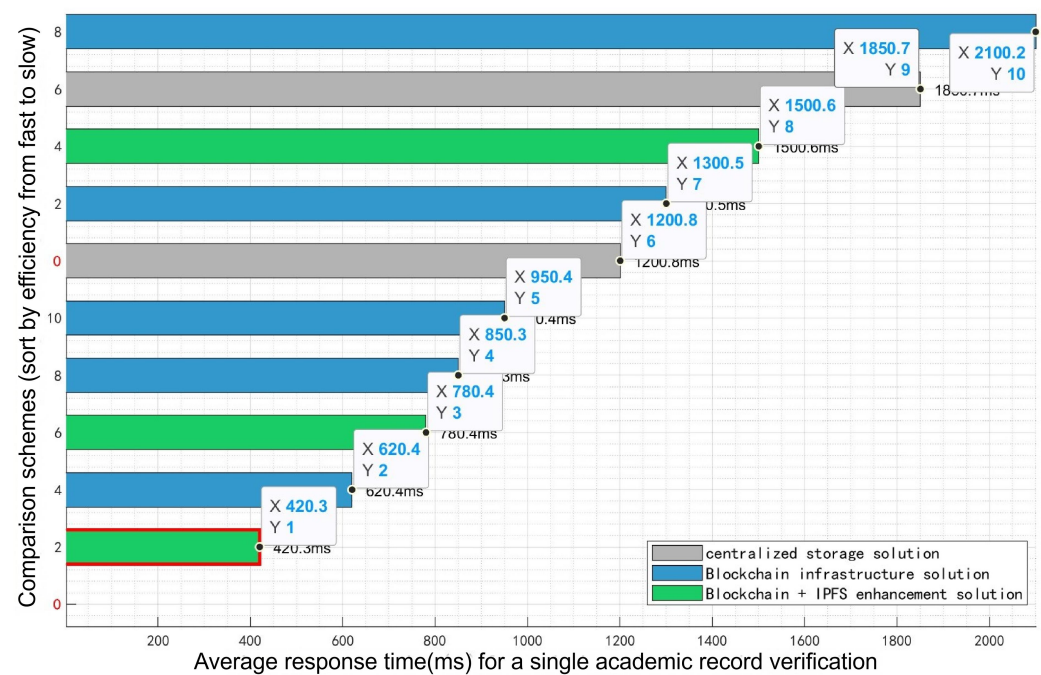


Figure 9. High-frequency verification efficiency test results graph, where Y-axis values represent the following: 10. Centralized DB+RBAC_Ministry of Education industry standard (GB/T 39784 [37]), 9. HBase+attribute encryption_Su et al. [12], 8. blockchain+homomorphic signature_Li et al. [13], 7. blockchain+basic ZKP_Fang et al. [11], 6. blockchain+zk-STARK_Saad Alahmari [16], 5. IPFS+static PRE_Sun et al. [18], 4. blockchain+dynamic PRE_Wang, S. [38], 3. Hyperledger+symmetric encryption_Guo et al. [1], 2. EOS+lightweight ZKP_education industry open-source solution, 1. Ethereum+optimized ZKP+PRE_this solution.

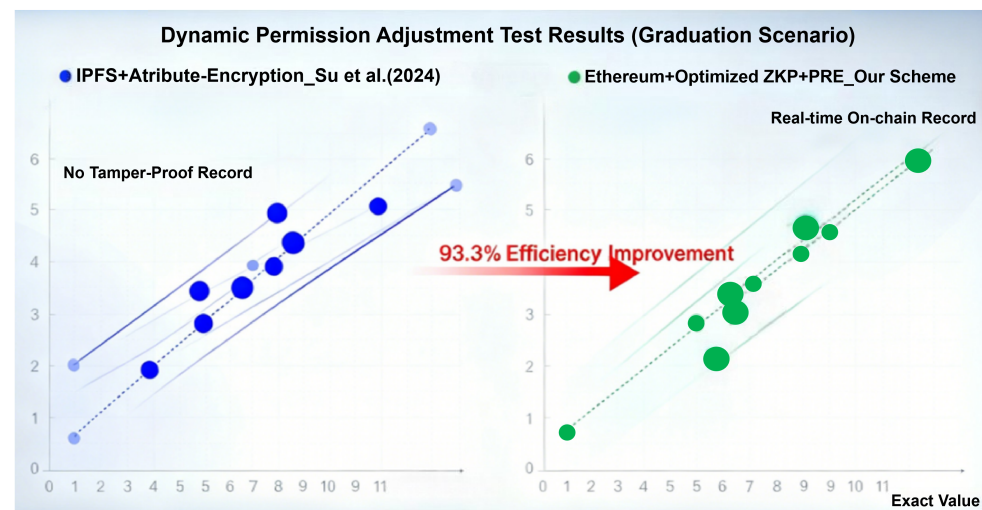


Figure 10. Dynamic permission adjustment test diagram Su et al. [12].

6.4. End-to-End Performance Evaluation

To comprehensively characterize the system-level feasibility of the proposed academic record-sharing framework, we evaluate two core performance metrics (latency and throughput) for the complete workflow (Request Submission → ZKP Proof Generation & Verification → PRE Re-encryption → Decryption) under a representative batch verification scenario. The test environment adopts distributed deployment: 10 Ethereum validator nodes, 60 IPFS sharded nodes, and a Kintex UltraScale KU115 FPGA accelerator. Concurrent requests are set to 10–200 to simulate real-world load fluctuations.

6.4.1. Latency Analysis

Latency is defined as the total time from request initiation to successful decryption of valid academic records. As illustrated in Figure 11, the proposed scheme maintains stable low latency across all concurrent loads: the average latency is 203.5 ms at 10 concurrent requests and only increases to 287 ms at 200 concurrent requests.

Compared with mainstream schemes:

- The IPFS+ABE scheme [18] exhibits 385 ms latency at 10 requests (47.2% higher than the proposed scheme) and 621 ms at 200 requests (61.4% growth);
- The Blockchain+zk-STARK scheme [16] shows 298 ms (31.6% higher) and 492 ms (52.1% growth);
- The centralized DB+RBAC scheme(GB/T 39784) has 285 ms (31.7% higher) and 498 ms (52.7% growth).

The latency stability of the proposed scheme benefits from FPGA-accelerated ZKP (accounting for 44.1% of total latency) and IPFS sharding, which effectively mitigate bottlenecks in proof generation and data retrieval. The sub-300 ms latency at high concurrency meets the requirements of real-time verification scenarios.

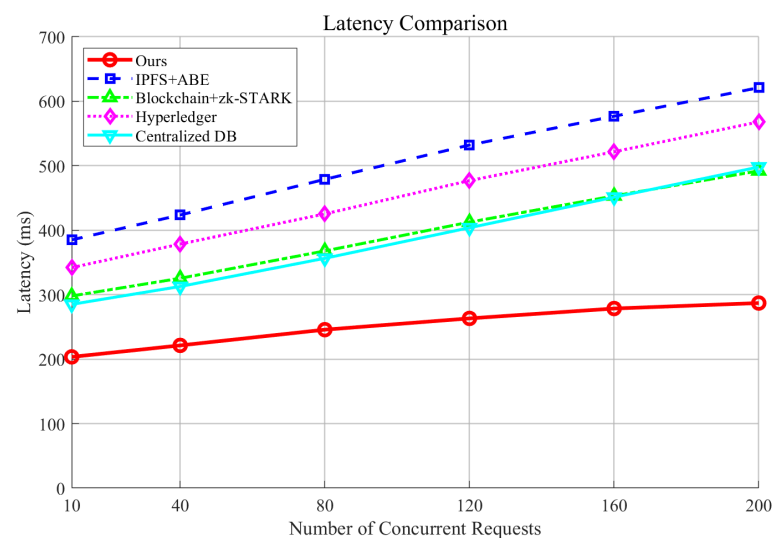


Figure 11. Latency variation with concurrent requests (10–200).

6.4.2. Throughput Analysis

Throughput (Transactions Per Second, TPS) reflects the system's large-scale processing capability. As shown in Figure 12, the proposed scheme achieves a peak throughput of 148 TPS at 200 concurrent requests, which is

- $2.3\times$ higher than the IPFS+ABE scheme;
- $1.8\times$ higher than the Blockchain+zk-STARK scheme;
- $3.1\times$ higher than the centralized DB+RBAC scheme.

When concurrent requests exceed 40, the throughput stabilizes above 120 TPS, enabling batch processing of 10,000 records in ~ 83 s and 500,000 records per hour—sufficient for educational authorities and large enterprises. The high throughput originates from two core optimizations:

1. Parallel processing of ZKP proof generation (FPGA multi-core architecture) and PRE re-encryption;
2. IPFS distributed storage reduces data retrieval latency, supporting efficient batch data access.

Notably, the throughput of the proposed scheme grows linearly with concurrent requests, demonstrating superior scalability compared to schemes with exponential performance degradation—this is critical for long-term system expansion.

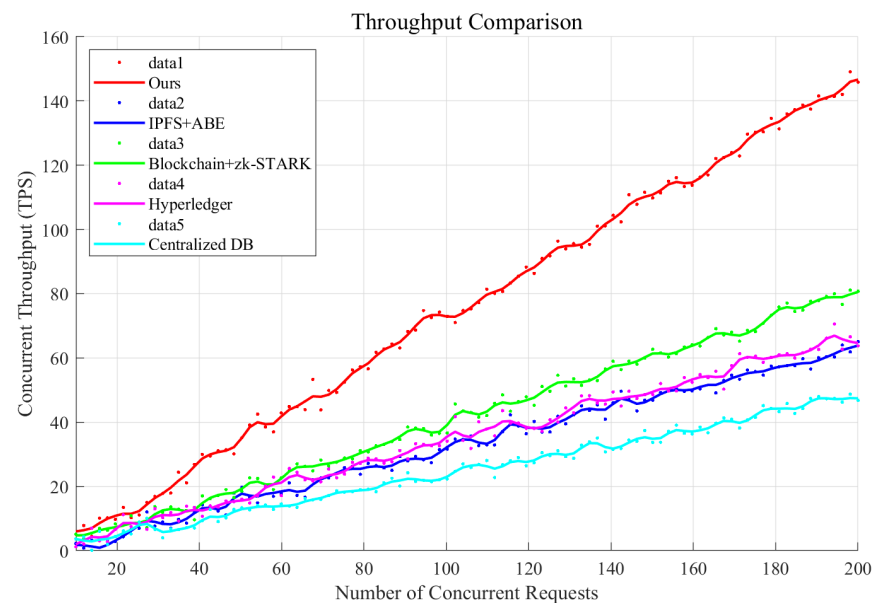


Figure 12. Throughput variation with concurrent requests (10–200).

7. Conclusions

This study proposes a dedicated sharing framework for SEARs integrating ZKP, PRE, EB, IPFS, and FPGA acceleration, addressing the core challenges of security, privacy, and efficiency in SEARs sharing. Quantitative results demonstrate that the optimized ZKP, combined with gate merging, constant folding, and FPGA acceleration, reduces the proof generation time to 90.47 ms and the verification time to 20.18 ms, supporting over 10,000 daily batch verifications with a single verification response time of 420.3 ms; this represents 80.1% and 50.6% higher efficiency than the blockchain+basic ZKP and blockchain+zk-STARK schemes. The PRE based on the LWE problem ensures a zero decryption success rate for unauthorized users and a dynamic permission adjustment response time of 35 ms, corresponding to a 93.3% efficiency improvement. The IPFS-blockchain architecture reduces the initial deployment cost to ≤ 8000 USD, annual operation and maintenance cost to $\leq 80,000$ USD, and communication overhead under high concurrency of 200 requests to 3.72 GB/h—47.2% lower than the IPFS+ABE scheme. The system achieves a peak throughput of 148 Transactions Per Second and maintains end-to-end latency between 203.5 ms and 287 ms, with success rates of 51% attacks, semi-honest proxy attacks, and identity forgery attacks all below 0.001% and the pass probability of forged proofs $\leq 2^{-256}$. These results validate the framework's superior performance in balancing security, privacy, and efficiency. The framework can be widely applied to cross-institutional academic certification, employer batch degree verification, long-term SEARs archiving, and regulatory auditing, providing a trusted technical foundation for the digital education ecosystem. Future extensions will focus on post-quantum security enhancement and educational big data collaborative analytics.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/fi18010047/s1>.

Author Contributions: Conceptualization, X.L. and M.T.; methodology, X.L.; software, X.L.; validation, X.L. and M.T.; formal analysis, X.L.; investigation, X.L.; resources, M.T.; data curation, X.L.;

writing—original draft preparation, X.L.; writing—review and editing, X.L. and M.T.; visualization, X.L.; supervision, M.T.; project administration, M.T.; funding acquisition, M.T.; experimental guidance, W.T. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Natural Science Foundation of Hunan Province, China (No. 2022JJ50153) and the 2019 Scientific Research Project of Hunan Provincial Department of Finance, China (No. XJCZ2019-15).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data supporting this study are included in the article and its Supplementary Materials.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Guo, J.; Zhao, K.; Liang, Z.; Min, K. Efficient and Secure EMR Storage and Sharing Scheme Based on Hyperledger Fabric and IPFS. *Appl. Sci.* **2024**, *14*, 5005. [\[CrossRef\]](#)
- Daraghmi, E.-Y.; Daraghmi, Y.-A.; Yuan, S.-M. UniChain: A Design of Blockchain-Based System for Electronic Academic Records Access and Permissions Management. *Appl. Sci.* **2019**, *9*, 4966. [\[CrossRef\]](#)
- Becke, M.; Padberg, J. Der Weg zur digitalen Arbeitsmappe: Digitales Prüfungswesen mit Zertifizierung. *arXiv* **2024**, arXiv:2408.09184. [\[CrossRef\]](#)
- Bawane, D.H.; Sudke, Y.S.; Pore, Y.N. Educational Record Authentication Using Decentralized Consortium Blockchain Technology. *Int. J. Res. Appl. Sci. Eng. Technol.* **2023**, *11*, 52379. [\[CrossRef\]](#)
- Dhinakaran, D.; Selvaraj, D.; Dharini, N.; Raja, S.E.; Priya, C.S.L. Towards a Novel Privacy-Preserving Distributed Multiparty Data Outsourcing Scheme for Cloud Computing with Quantum Key Distribution. *arXiv* **2024**, arXiv:2407.18923.
- Zhang, R.; Xue, R.; Liu, L. Security and Privacy on Blockchain. *ACM Comput. Surv.* **2019**, *52*, 51. [\[CrossRef\]](#)
- Ma, W.; Wei, X.; Wang, L. A Security-Oriented Data-Sharing Scheme Based on Blockchain. *Appl. Sci.* **2024**, *14*, 6940. [\[CrossRef\]](#)
- Qi, F.; He, D.B.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58. [\[CrossRef\]](#)
- Gabbay, M.J. Decentralised collaborative action: Cryptoeconomics in space. *arXiv* **2025**, arXiv:2504.12493. [\[CrossRef\]](#)
- Zhang, J.Y.; Guo, R.X.; Shi, Y.F.; Tang, W.T. An anti-impersonation attack electronic health record sharing scheme based on proxy re-encryption and blockchain. *Math. Biosci. Eng.* **2024**, *21*, 6167–6189. [\[CrossRef\]](#)
- Fang, Y.; Bi, W.B.; Cao, N.; Luo, J.; An, D.T.; Ding, L.Q.; Higgs, R. Research on Multi-Blockchain Electronic Archives Sharing Model. *Comput. Mater. Contin.* **2023**, *76*, 3921–3931. [\[CrossRef\]](#)
- Su, H.Q.; Li, J.W.; Guo, L.; Wang, W.S.; Yang, Y.J.; Wen, Y.; Li, K.; Mo, P.Y. Massive Data HBase Storage Method for Electronic Archive Management. *Int. J. Netw. Manag.* **2024**, *35*, e2308. [\[CrossRef\]](#)
- Li, J.W.; Su, H.Q.; Guo, L.; Wang, W.S.; Yang, Y.J.; Wen, Y.; Li, K.; Mo, P.Y. Security Protection Method for Electronic Archives Based on Homomorphic Aggregation Signature Scheme in Mobile Network. *Int. J. Netw. Manag.* **2024**, *35*, e2316. [\[CrossRef\]](#)
- Liu, X.; Chen, W.T.; Peng, L.; Luo, D.; Jia, L.K.; Xu, G.; Chen, X.B.; Liu, X.M. Secure computation protocol of Chebyshev distance under the malicious model. *Sci. Rep.* **2024**, *14*, 17115. [\[CrossRef\]](#)
- Al-Khasawneh, M.A.; Faheem, M.; Alarood, A.A.; Habibullah, S.; Alzahrani, A. A secure blockchain framework for healthcare records management systems. *Healthc. Technol. Lett.* **2024**, *11*, 461–470. [\[CrossRef\]](#)
- Alahmari, S.; Alshardan, A.; Al-Wesabi, F.N.; Sorour, S.; Alghushairy, O.; Alsini, R.; Khadidos, A.O.; Al Duhayyim, M. A decentralized and privacy-preserving framework for electronic health records using blockchain. *Alex. Eng. J.* **2025**, *126*, 196–203. [\[CrossRef\]](#)
- Guo, C.J.; You, L.; Li, X.Y.; Hu, G.G.; Wang, S.G.; Cao, C.T. A novel biometric authentication scheme with privacy protection based on SVM and ZKP. *Comput. Secur.* **2024**, *144*, 103995. [\[CrossRef\]](#)
- Sun, J.; Yao, X.M.; Wang, S.P.; Wu, Y. Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS. *IEEE Access* **2020**, *8*, 59389–59401. [\[CrossRef\]](#)
- Wang, B.; Tian, Z.; Liu, X.R.; Xia, Y.J.; She, W.; Liu, W. A multi-center federated learning mechanism based on consortium blockchain for data secure sharing. *Knowl.-Based Syst.* **2025**, *310*, 112962. [\[CrossRef\]](#)
- Wang, Y.; Ismail, E.S. A Review on the Advances, Applications, and Future Prospects of Post-Quantum Cryptography in Blockchain and IoT. *IEEE Access* **2025**, *13*, 112962–112977. [\[CrossRef\]](#)

21. Kan, J.; Zhang, J.; Liu, D.; Huang, X. Proxy Re-Encryption Scheme for Decentralized Storage Networks. *Appl. Sci.* **2022**, *12*, 4260. [CrossRef]
22. Sun, X.Q.; Yu, F.R.; Zhang, P.; Sun, Z.W.; Xie, W.X.; Peng, X. A Survey on Zero-Knowledge Proof in Blockchain. *IEEE Netw.* **2021**, *35*, 198–205. [CrossRef]
23. Bünz, B.; Bootle, J.; Boneh, D.; Poelstra, A.; Wuille, P.; Maxwell, G. Bulletproofs: Short Proofs for Confidential Transactions and More. *IEEE Symp. Secur. Priv.* **2018**. Available online: <https://eprint.iacr.org/2017/1066.pdf> (accessed on 15 November 2025).
24. Weng, K.K.; Yang, K.; Katz, J.; Wang, X. Wolverine: Fast, Scalable, and Communication-Efficient Zero-Knowledge Proofs for Boolean and Arithmetic Circuits. *IEEE Symp. Secur. Priv.* **2021**. Available online: <http://eprint.iacr.org/2020/1169.pdf> (accessed on 15 November 2025).
25. Groth, J. On the Size of Pairing-Based Non-Interactive Arguments. *ASIACRYPT* **2016**. Available online: <https://eprint.iacr.org/2016/260.pdf> (accessed on 16 November 2025).
26. Ben-Sasson, E.; Bentov, I.; Horesh, Y.; Riabzev, M. Scalable, Transparent, and Post-Quantum Secure Computational Integrity. *CRYPTO* **2018**. Available online: <https://eprint.iacr.org/2018/046.pdf> (accessed on 16 November 2025).
27. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC '05), Baltimore, MD, USA, 22–24 May 2005; Association for Computing Machinery: New York, NY, USA, 2005; pp. 84–93. [CrossRef]
28. Benet, J. IPFS—Content Addressed, Versioned, P2P File System. *arXiv* **2014**, arXiv:1407.3561.
29. Goldwasser, S.; Micali, S.; Rackoff, C. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.* **1989**, *18*, 186–208. [CrossRef]
30. Damascevicius, R.; Bacanin, N.; Nayyar, A. Blockchain technology for a trustworthy social cblackit system: Implementation and enforcement perspectives. *Clust. Comput.* **2025**, *28*, 162. [CrossRef]
31. Guo, C.; Peng, W.J.; Wu, J.; Fang, Y.X.; Ye, K.K.; Xin, Y.S. A blockchain-based proxy re-encryption scheme with conditional privacy protection and auditability. *China Commun.* **2024**, *21*, 267–277. [CrossRef]
32. Wu, G.F.; Wang, H.P.; Lai, X.; Wang, M.M.; He, D.J.; Chan, S. A comprehensive survey of smart contract security: State of the art and research directions. *J. Netw. Comput. Appl.* **2024**, *226*, 103882. [CrossRef]
33. Chen, C.; Li, Y.C.; Wu, Z.P.; Mai, C.Y.; Liu, Y.M.; Hu, Y.M.; Kang, J.W.; Zheng, Z.B. Privacy computing meets metaverse: Necessity, taxonomy and challenges. *Ad Hoc Netw.* **2024**, *158*, 103457. [CrossRef]
34. FIPS PUB 180-4; Federal Information Processing Standards Publication; Secure Hash Standard (SHS). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015. Available online: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf> (accessed on 18 November 2025).
35. Yang, Y.K.; Lu, Z.Y.; Zeng, J.W.; Liu, X.G.; Qian, X.H.; Yu, Z.B. Falic: An FPGA-Based Multi-Scalar Multiplication Accelerator for Zero-Knowledge Proof. *IEEE Trans. Comput.* **2024**, *73*, 2791–2804. [CrossRef]
36. Yang, J.C.; Li, L.; Gu, Y.W.; Wu, H.Q. Fast Authenticated and Interoperable Multimedia Healthcare Data over Hybrid-Storage Blockchains. *arXiv* **2025**, arXiv:2510.13318. [CrossRef]
37. GB/T 39784-2021; General Functional Requirements for Electronic Archive Management Systems. State Administration for Market Regulation: Beijing, China, 2021.
38. Wang, S.; Luo, N.; Xing, B. Blockchain-based proxy re-encryption access control method for biological risk privacy protection of agricultural products. *Sci. Rep.* **2024**, *14*, 20048. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.