

A Block Chain-Based Framework for Secure and Authentic Academic Document Verification

Mrs. Jahanara Begum¹, Thatipamula Hasini², Sangeneeni Abhinav³, Thammisetty RatanVeer⁴,
Vakati Sai Prashanth Reddy⁵

¹Assistant Professor, Dept. of Computer Science and Engineering, HITAM, Telangana, India

²³⁴⁵Student, Dept. of Computer Science and Engineering, HITAM, Telangana, India

Abstract - Academic certificates are essential for verifying qualifications, but traditional verification is often time-consuming and vulnerable to forgery. This paper presents an Academic Authenticity Validator that combines blockchain, SHA-256 hashing, and secure authentication for academic document verification. The system uses React.js, Node.js, Express.js, and MongoDB, with JWT for authentication and Multer for document uploads. A unique SHA-256 hash is generated for each document and stored through a Solidity smart contract on the blockchain. During verification, the document is hashed again and compared with the stored hash to detect tampering and confirm authenticity. The proposed system provides a secure, transparent, and efficient approach to academic credential verification.

Key Words: Blockchain, Document Authentication, Cryptographic Hashing, Smart Contracts, Digital Credentials, Data Integrity, Certificate Security, Fraud Detection, Web Application, Decentralized Verification.

1.INTRODUCTION

Academic documents such as degree certificates, mark sheets, transcripts, and other educational credentials are essential for higher education, employment, and professional qualifications. However, digital editing tools have increased the risk of modification, duplication, and forgery. Traditional verification depends on manual communication between organizations and issuing institutions, making the process time-consuming and creating concerns related to data integrity, unauthorized modification, and centralized control.

Blockchain technology provides an immutable and transparent approach by storing a cryptographic hash of the document through a smart contract instead of storing the complete document.

The proposed Academic Authenticity Validator uses blockchain and cryptographic hashing to provide secure academic document verification. Authorized users can upload documents, generate SHA-256 hashes, store certificate information through a Solidity smart contract,

and verify documents by comparing newly generated hashes with blockchain records. The application also incorporates JWT-based authentication, role-based access, MongoDB storage, and a React.js interface to provide a secure and efficient verification platform.

1.1 Background of Academic Document Verification

Academic credential verification is important for educational institutions, employers, government organizations, and other authorized bodies to ensure that certificates and educational records are genuine and unmodified. Conventional methods usually require contacting the issuing institution or checking centralized records, which can cause delays and administrative workload. Digital documents also increase the risk of undetected modification and tampering. Blockchain provides a distributed and tamper-resistant method for maintaining verification records.

By storing the cryptographic hash of an academic document on the blockchain, the stored hash can be used as a reference when the same document is submitted later. Comparing the two hashes helps determine whether the document matches the originally registered record.

1.2 Need for Secure Academic Credential Verification

The growing use of digital academic records has increased the need for verification systems that are secure, reliable, and easy to use. Manual verification can become difficult when institutions, employers, and other organizations have to process a large number of certificates. Delays in verification can also affect students and organizations that depend on timely confirmation of academic qualifications. The proposed system addresses these challenges by combining SHA-256 hashing with blockchain-based storage. A SHA-256 hash acts as a digital fingerprint of the uploaded document, while the Solidity smart contract stores the certificate information and its corresponding hash. When a document is submitted for verification, the system generates a new hash and compares it with the blockchain record. If both values match, the document corresponds to the registered record. JWT-based

authentication is also used to restrict protected operations to authorized users.

2. LITERATURE REVIEW

Several studies have explored blockchain, cryptographic hashing, and digital technologies to improve the security, integrity, and efficiency of academic credential verification.

Blockchain-based academic credential verification has received considerable attention because academic certificates need a reliable method for proving their authenticity. Previous research has investigated the use of blockchain networks, smart contracts, cryptographic techniques, decentralized storage, and access-control mechanisms for managing academic credentials.

Existing studies show that blockchain can provide transparency and help protect certificate records from unauthorized changes. However, different systems focus on different aspects of the verification process.

Some mainly concentrate on blockchain storage, while others focus on credential management, access control, or decentralized verification. This creates an opportunity to develop a complete web-based application that combines these technologies into a single verification workflow.

2.1 Blockchain-Based Academic Certificate Verification

Blockchain can be used to maintain academic credential records in a distributed and tamper-resistant manner. Research in this area shows that storing certificate information or its cryptographic representation on a blockchain can reduce dependence on centralized verification authorities.

Smart contracts can also simplify the process by defining how certificate information is stored and retrieved. Authorized users can submit certificate information and compare it with records maintained on the blockchain. However, some existing approaches focus mainly on blockchain storage and verification and provide limited support for practical application features such as user authentication, document uploading, and web-based verification.

2.2 Cryptographic Hashing for Document Integrity

Cryptographic hashing provides a simple way to represent a digital document using a fixed-length value. In the proposed system, SHA-256 is used to generate this value, which acts as a digital fingerprint of the uploaded

certificate.

When a certificate is submitted for verification, the system generates its SHA-256 hash again. If the new hash matches the hash stored on the blockchain, the submitted document corresponds to the registered record. If the document has been modified, even slightly, its hash will be different.

Therefore, cryptographic hashing provides a reliable way to check document integrity and identify unauthorized modifications.

2.3 Smart Contracts for Academic Credential Management

Smart contracts are programs stored on a blockchain that execute predefined operations according to their programmed logic. They can be used to store and retrieve academic certificate information in a consistent manner.

In the proposed system, a Solidity smart contract stores details such as the student name, course, certificate hash, issuer, and timestamp. The contract provides functions for registering and retrieving certificate information. This reduces the need for manual handling of verification records and provides a blockchain-based reference that can be used during future verification.

2.4 Digital and QR-Based Certificate Verification

QR codes can provide a convenient way to access certificate verification information. A QR code can contain a verification identifier or URL that directs a user to the appropriate verification process. This can make verification easier because users do not have to manually enter certificate details.

QR-based verification can therefore be used as an additional feature alongside blockchain-based verification. In the current project, this feature can be considered as an extension of the core hash-based verification process.

2.5 AI-Based Document Analysis

Artificial Intelligence can provide additional capabilities for analyzing academic documents. Techniques such as Optical Character Recognition (OCR), plagiarism detection, and document analysis can be used to examine document content and identify suspicious characteristics.

In the proposed project, AI-based analysis is considered an additional validation layer. Blockchain and SHA-256 hashing are mainly responsible for checking document integrity, while AI can potentially be used to examine the content and structure of documents. This can extend the system beyond simple hash comparison in future

development.

2.6 Research Gap

Existing research demonstrates the usefulness of blockchain, cryptographic hashing, and smart contracts for academic credential verification. However, some practical limitations still exist.

Many systems focus mainly on storing and verifying credentials on the blockchain without combining authentication, document uploading, and application-level access control into one complete web application.

Some approaches also provide limited support for convenient verification methods and additional document analysis. In addition, several studies are conceptual or review-based rather than demonstrating a complete end-to-end application.

The proposed Academic Authenticity Validator addresses these gaps by combining document upload, SHA-256 hashing, blockchain-based certificate records, Solidity smart contracts, JWT authentication, MongoDB, and a web-based verification workflow. The aim is to provide a practical system for secure and efficient academic document verification.

3.METHODOLOGY

The proposed system follows a system-development approach that combines a web application with blockchain-based verification. The application uses React.js for the frontend, Node.js and Express.js for backend services, MongoDB for application data, JWT for authentication, SHA-256 for document hashing, and a Solidity smart contract developed using Hardhat.

The overall verification process follows these steps:
User Authentication → Document Upload → SHA-256 Hash Generation → Database Processing → Blockchain Storage → Verification Request → Hash Comparison → Verification Result

The main purpose of this workflow is to maintain the integrity of academic documents while providing an easy and secure verification process.

3.1 Proposed System

The Academic Authenticity Validator is designed as a web-based platform for registering and verifying academic documents. Users first authenticate themselves before accessing protected operations. When an academic certificate is uploaded, the backend processes the file and generates its SHA-256 hash. The certificate details and hash are then associated with the certificate record. The

smart contract stores the required certificate information and hash on the blockchain.

When a certificate needs to be verified, the submitted document is processed again and a new hash is generated. This hash is compared with the corresponding blockchain record. If both hashes match, the document corresponds to the registered record. If they are different, the submitted document does not match the stored record.

3.2 System Architecture

The system is organized into four main layers: the presentation layer, application layer, database layer, and blockchain layer.

The presentation layer is developed using React.js and provides interfaces for user registration, login, certificate upload, and verification. The application layer uses Node.js and Express.js to handle API requests, authentication, document processing, and communication with other components. MongoDB is used to store application-related information such as user details and certificate metadata. The blockchain layer uses a Solidity smart contract to maintain certificate information and document hashes.

The major technologies used in the system are shown below.

Component	Technology
Frontend	React.js
Backend	Node.js, Express.js
Database	MongoDB
Authentication	JWT
File Upload	Multer
Hashing	SHA - 256
Smart Contract	Solidity
Blockchain Development	Hardhat
Blockchain Network	HardhatLocalNetwork
Version Control	Git and GitHub

3.3 User Authentication and Role-Based Access

Authentication is used to protect the system from unauthorized access. The application uses JSON Web Token (JWT) based authentication.

During registration, the user's information is stored in the database after secure password processing. When the user logs in, the submitted credentials are checked and a JWT is generated for successful authentication. Protected backend routes verify this token before allowing access to restricted operations.

Role-based access can also be used to provide different permissions to users such as administrators, institutions, students, and authorized verifiers. This provides an additional layer of security for certificate management and verification.

3.4 Academic Document Upload

Authorized users can upload academic certificates through the web application. Multer is used in the backend to handle multipart file uploads.

Along with the uploaded document, relevant information such as the student's name and course is recorded. The backend creates a certificate record and associates the uploaded document with the authenticated user.

This upload process forms the starting point for generating the document hash and registering the certificate for blockchain-based verification.

3.5 SHA-256 Hash Generation

After a certificate is uploaded, a SHA-256 hash is generated from the document. The hash can be represented as: $H = \text{SHA-256}(D)$ where D represents the uploaded document and H represents its generated hash value.

The generated hash acts as a digital fingerprint of the document. If the contents of the document are changed, the resulting hash will also change. This makes it possible to identify whether a submitted document matches the previously registered document.

3.6 Blockchain and Smart Contract Integration

The generated certificate hash and important certificate details are stored using a Solidity smart contract.

The certificate record contains information such as:

- Student name
- Course
- Certificate hash
- Issuer
- Timestamp

The issue Certificate function is used to register certificate information, while the verify Certificate function is used to retrieve the stored information for verification.

Hardhat is used as the blockchain development framework. The smart contract is deployed on the configured local blockchain network, and the backend communicates with the deployed contract to store and retrieve certificate information.

3.7 QR Code generation

A QR-based verification mechanism can be added to an academic certificate to make the verification process easier. The QR code can contain a verification identifier or URL that directs the user to the appropriate verification page.

This can be useful for employers, educational institutions, and other authorized organizations because they can access the verification process without manually entering certificate information. In the current system, QR-based verification is considered an additional feature that can be integrated with the main blockchain verification process.

3.8 Document Verification

Document verification is performed by generating a new SHA-256 hash from the submitted academic document and comparing it with the hash stored on the blockchain.

The verification process can be represented as:

Submitted Document → SHA-256 Hash → Blockchain Hash Comparison → Verification Result

If:

Hash Submitted = Hash Blockchain

the document matches the registered record.

If:

Hash Submitted ≠ Hash Blockchain

submitted document does not match the registered record and may have been modified or may belong to a different certificate. This provides an automated way of checking the integrity of an academic document.

3.9 Database and Backend

MongoDB is used to manage application-level information such as user and certificate records. The Node.js and Express.js backend provides the APIs through which the frontend communicates with the database.

The backend manages user registration, login, protected routes, certificate uploads, and certificate-related information. Authentication middleware is used to protect operations that require a logged-in user. Separating the database and blockchain responsibilities makes the system more modular. MongoDB handles normal application data, while the blockchain maintains the certificate information required for verification.

3.10 AI-Based Validation

AI-based document validation is considered an additional

feature that can improve document analysis. Possible techniques include Optical Character Recognition (OCR), plagiarism detection, and document forgery analysis. OCR can be used to extract text from scanned documents, while document analysis can examine their structure and content. Plagiarism detection can also be considered for suitable academic documents such as research papers.

The AI component is intended to complement blockchain verification rather than replace it. Blockchain and hashing can verify document integrity, while AI can provide additional content-level analysis in future development.

3.11 Verification Workflow

The complete verification workflow consists of the following steps:

1. The authorized user logs into the application.
2. The academic document is uploaded.
3. The backend processes the uploaded file.
4. A SHA-256 hash is generated.
5. Certificate information and the hash are registered through the smart contract.
6. The certificate can optionally be associated with a QR-based verification mechanism.
7. A verifier submits the certificate for verification.
8. The system generates a new SHA-256 hash.
9. The new hash is compared with the blockchain record.
10. The system displays the corresponding verification result.

This workflow provides a simple and automated method for checking whether an academic document matches its registered blockchain record.

4. IMPLEMENTATION AND RESULTS

The Academic Authenticity Validator is being developed as a full-stack web application using:

React.js, Node.js, Express.js, MongoDB, JWT, Multer, Solidity, and Hardhat. The development is divided into several components, including the frontend, backend, database, authentication, document processing, and blockchain integration.

The implementation includes the development of the React.js frontend, Express.js backend, MongoDB connectivity, authentication routes, certificate upload functionality, smart contract development, and blockchain integration.

4.1 System Implementation

The frontend is developed using React.js and provides the interface for user authentication and certificate-related operations. The backend uses Node.js and Express.js to provide APIs for user registration, login, protected profile access, and certificate upload.

MongoDB is used for application data storage, while Multer handles certificate file uploads. JWT is used to authenticate requests to protected backend routes.

The blockchain component contains the AcademicCertificate Solidity smart contract. The contract stores important certificate information along with the corresponding document hash and provides functions for issuing and verifying certificate records.

4.2 User Authentication Results

The authentication module supports user registration and login.

During registration, the user's password is securely processed before being stored in the database. After successful login, the system generates a JWT for the authenticated user.

The authentication middleware checks the JWT before allowing access to protected routes. This prevents users who are not authenticated from directly accessing restricted certificate operations.

4.3 Document Upload and Hash Generation

The certificate upload module accepts academic documents through the backend API. Multer is used to process the uploaded files, and certificate information is associated with the authenticated user.

After the document is received, its SHA-256 hash can be generated to create a digital fingerprint. This hash acts as the reference value for later verification. If the contents of the document are changed, a different hash will be produced, allowing the system to identify a mismatch.

4.4 Blockchain Storage and Verification

The Solidity smart contract provides the blockchain component for storing academic certificate information. A certificate record contains the student name, course, certificate hash, issuer, and timestamp.

The issueCertificate function is used to register certificate information, while the verifyCertificate function retrieves the stored record. Hardhat is used for smart contract

development and deployment on the configured local blockchain network.

The backend is configured to communicate with the deployed smart contract, allowing certificate information and hashes to be used during the verification process.

4.5 Database Integration

MongoDB is used to store application-level information such as user details and certificate - related data. The backend establishes the database connection and uses models to manage these records.

The database and blockchain serve different purposes in the system. MongoDB handles regular application operations, while the blockchain provides a tamper-resistant reference for certificate verification data.

This avoids the need to store the complete application data directly on the blockchain.

4.6 Verification Results

The verification process is based on comparing the SHA-256 hash of the submitted document with the corresponding hash stored on the blockchain.

If an unchanged certificate produces the same hash as the registered record, the document matches the stored certificate information. If the document has been modified, the generated hash will be different from the stored value. This provides a straightforward method for checking the integrity of submitted academic documents.

4.7 Security Analysis

The system combines several security mechanisms to protect academic certificate information. SHA-256 provides a cryptographic representation of each document, while blockchain storage provides a tamper-resistant reference for the registered hash. Smart contracts help manage certificate records according to predefined rules, and JWT authentication protects restricted application routes. MongoDB is used for regular application data, while blockchain is used for certificate verification information.

By keeping the complete document separate from the blockchain and storing its hash instead, the system can maintain a verifiable reference without placing the entire academic document on the blockchain.

5. DISCUSSION

The Academic Authenticity Validator combines authentication, document management, cryptographic

hashing, and blockchain technology to address some of the difficulties associated with conventional academic document verification. In traditional verification, an organization may need to contact the institution that issued a certificate before confirming its authenticity. The proposed system provides a stored cryptographic reference that can be used to check the submitted document against the registered blockchain record.

SHA-256 plays an important role because the system does not need to compare the entire document directly. Instead, it compares the digital fingerprints generated from the documents. Even a small change in the document can produce a different hash, making it possible to identify a mismatch.

JWT authentication adds security at the application level by restricting protected operations to authenticated users. MongoDB and blockchain are also used for different purposes. MongoDB manages regular application information, while blockchain maintains the certificate information required for verification.

5.1 Analysis of the Proposed System

The proposed system brings together several technologies to create a practical academic document verification platform.

The main benefits of the approach include:

Reduced dependence on manual certificate verification.
Hash-based checking of document integrity.

- Tamper-resistant certificate records through blockchain.
- Secure authentication using JWT.
- Application data management using MongoDB.
- Automated document verification through hash comparison.
- A foundation for future QR-based verification.
- Scope for adding AI-based document analysis in future.
- The combination of these components provides a practical foundation for secure academic credential verification.

5.2 Comparison with Existing Approaches

Existing blockchain-based academic credential systems have demonstrated the usefulness of blockchain for certificate issuance and verification.

However, implementations differ in areas such as authentication, document management, verification methods, and additional analysis features.

The proposed system focuses on bringing several of these components together within a single web application. Instead of using blockchain only as a storage mechanism, it combines blockchain with a conventional full-stack architecture. SHA-256 is used for document-level integrity checking, while the Solidity smart contract manages blockchain-based certificate information. JWT and MongoDB support the application-level requirements of the system.

5.3 Practical Significance of the Proposed System

The proposed system can be useful for educational institutions, employers, universities, and other authorized organizations that need to verify academic credentials.

It can help with:

- Faster checking of academic documents.
- Reduction of manual verification work.
- Detection of modified documents through hash comparison.
- Secure management of certificate information.
- Improved transparency of verification records.
- Reduced need for repeated communication with issuing institutions.

The system therefore demonstrates how blockchain and cryptographic hashing can be combined with a web application to support academic document verification.

6. CONCLUSION

Academic document verification is an important process for educational institutions, employers, universities, and other organizations. Traditional verification methods can require significant manual effort and may take time, especially when a large number of certificates need to be checked.

This paper presented the **Academic Authenticity Validator**, a blockchain-based approach for academic document verification. The system uses SHA-256 hashing to create a digital fingerprint of an academic document and a Solidity smart contract to maintain certificate information and its corresponding hash on the blockchain.

The application combines React.js for the frontend, Node.js and Express.js for backend services, MongoDB for application data, JWT for authentication, Multer for document uploads, and Hardhat and Solidity for blockchain development. During verification, the submitted document is hashed again and the generated value is compared with the registered blockchain record.

The combination of these technologies provides a practical approach for improving document integrity and reducing dependence on manual verification. The system also provides a foundation for adding further features as the project develops.

6.1 Future Scope

The system can be extended in several ways in the future. AI-based document analysis could be added to identify suspicious document structures, extract information using OCR, and support plagiarism detection for suitable academic documents.

The blockchain component could also be moved from the current local development environment to a suitable public or permissioned blockchain network. Decentralized storage technologies such as IPFS could be explored for storing academic documents.

Other possible improvements include certificate revocation, verification history, institution-level dashboards, automated verification reports, and mobile-based QR verification. The system could also be tested with a larger collection of academic documents to evaluate its performance under different verification conditions.

REFERENCES

- 1) K. Tagore and S. Jayalakshmi, "The Impact of the Blockchain on Academic Certificate Verification System – Review," EAI Endorsed Transactions on Energy Web, 2021.
- 2) A. S. Akinnifesi and J. M. Balogun, "Design and Implementation of a Blockchain-Based Certificate Verification System for Secure Academic Credential Authentication," University of Ibadan Journal of Science and Logics in ICT Research, 2025.
- 3) "Decentralized Certificate Issuance and Verification System Using Ethereum Blockchain Technology," Journal of Network and Computer Applications, 2025.
- 4) S. B. Pawar, G. N. Shevkar, D. R. Shinde, G. A. Somase, and P. S. Lahane, "CredVerify: Implementation and Validation of an Academic Certificate Verification Framework Using Blockchain," 2026.
- 5) A. Rustemi, F. Dalipi, V. Atanasovski, and A. Risteski, "Systematic Review of Blockchain-Based Certificate Verification Systems and Smart Contracts," 2023.