IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

Blockchain Driven Multi Tier Academic Credential Validation System

G. Priyadharshini M.E.[1], Prakash B[2], Logesh P[3]

Assistant Professor, CSE (Cyber Security) & Dhanalakshmi Srinivasan College of Engineering & Technology, India[1]

CSE (Cyber Security) & Dhanalakshmi Srinivasan College of Engineering & Technology, India[2]

CSE (Cyber Security) & Dhanalakshmi Srinivasan College of Engineering & Technology, India[3]

Abstract—The rise of fraudulent academic certificates has led to serious real-world issues, such as the use of fake degree certificates for jobs and visas. It also causes noteworthy disruptions in the hiring process. Although trust in institutions has been reduced, some research has been conducted to validate the authenticity of certificates. Blockchain-based verification systems focus on storing and validating certificates but often fail to provide integrated issuer authentication, tamper-proof privacy protection, and low-cost solutions. In this proposed work, research is done on an integrated, comprehensive, decentralised framework for certificate verification that together validates both the certificate and the issuer's authority. In addition, it protects the certificate from being tampered with. In addition, hash function mapping was employed for faster searching. The proposed solution is experimentally validated by creating a structure that incorporates zero-knowledge proofs (ZKPs) to safeguard data, self-destructing smart contracts to exclude replay attacks, and post-quantum lattice-based signatures to ensure security versus failures. In addition, a hash-based indexing method is used to accelerate certificate searches while reducing storage demands. In experimental analysis, measure the performance of proposed work in private. The Ethereum blockchain indicates that the proposed approach attains low transaction and gas costs, fast retrieval times, and high verification compared to previous research techniques. This overall analysis achieves 94% of a scalable and privacy-preserving solution for authenticating academic credentials in real-world scenarios.

Keywords: Block chain, Zero-knowledge proofs (ZKPs), Ethereum, Tamper-Proof, post-quantum lattice, hash-based indexing, smart contract.

I. INTRODUCTION

In the digital era, managing certificates, especially making sure they are tamper resistant and have integrity, is critical. As well, academic credentials are essential for employment, further education, work, and visas. The rule of these certificates is essential for preserving trust between organizations, employees, and authorities since they validate a person's educational history and ability. However, the advancement of digitalizing certificates Although centralized storage systems are used, distribution and storage have also increased[1].The worst aspect is that storage systems are easy targets for cyberattacks when they are employed for things that lead to a discernible increase in academic credentials that have been falsified or altered. These dishonest tactics have hampered hiring procedures and damaged institutional trust, resulting in financial [2].

Conventional certificate management systems mostly depend on centralized or semi-centralized infrastructures that are kept up to date by reputable third-party authorities or educational institutions.While public-key cryptography and digital signatures offer a low amount of safety, centralized control produces single points of failure that are vulnerable to ransomware, phishing, and SQLi attacks, account compromise, and data breaches [3]. Academic documents are exposed to illegal access and modification, as per cybersecurity surveys, making educational institutions one of the most targeted sections[4]. The effect of credential fraud extends beyond financial loss because individuals with fake credentials may lack the necessary proficiency, which could be risky in industries where safety is essential.

These difficulties are made worse by the physical and document-based certificate verification techniques that are still in use in many locations.These techniques are unsuitable for a variety of urgent verification applications because they are costly, time-consuming, and prone to human mistake. Institutions and businesses often have to validate thousands of academic degrees quickly since globalization encourages cross-border education and employment mobility.Conventional verification methods find it difficult to scale effectively due to these limitations, which raises the risk of fraud and delays operations [5].

Blockchain technology has become a trustworthy way of managing all types of digital content because it is a peer-to-peer network that has no single point of failure, records transactions. Blocks can be added or changed by consensus, but they are virtually impossible to erase once added. Therefore, blockchain technology provides the necessary ability to have an immutable record of transactions and keep track of the original creator of any content stored on blockchains.As such,

DOI: 10.17148/IJARCCE.2026.15553

blockchain technology has been utilized in many industries, including eCommerce, Internet-of-Things (IoT), Digital Identity, and Data Provenance / Traceability solutions[6].

Recent research has focused on developing academic credential verification systems on blockchain, where academic certificates or other cryptographic hash values are stored within the blockchain to ensure integrity and traceability. These solutions have improved model attacks on the verification systems through increased resistance to forgery; however, the majority of existing solutions only cover certain aspects of verification. More specifically, most academic credential verification systems do not incorporate an authentication mechanism for verifying the issuer of a credential. Therefore, the assumption that an issuer is legitimate is the default for most of the currently available credential verification systems.[7] Additionally, existing verification solutions have many technical limitations related to scalability, high transaction fees, slow retrieval times for certificates, and significant on-chain storage requirements limiting the practical implementation of these solutions. Another crucial issue with blockchain-based credential validation systems is privacy preservation. Sensitive personal information that shouldn't be disclosed during basic verification is sometimes found on academic credentials. Such systems run the danger of going against user privacy expectations and data protection concepts in the absence of suitable protections. Zero-knowledge proofs (ZKPs) are successful at providing privacy-preserving authentication and data verification without disclosing sensitive student information, according to recent studies[8]. However, current ZKP-based methods are not specifically tailored to integrated academic credential verification and are primarily intended for off-chain data flows, IoT networks, or common authentication.

Additionally, if implemented improperly, Smart Contracts are vulnerable to Replay Attacks and other known vulnerabilities that could compromise the security and reliability of the system. Furthermore, there is also the overlooked issue of "Cryptographic Longevity", where conventional Cryptographic Primitives (CRPD) may not be able to withstand the emergence of Quantum Computers. As such, there is increasing demand for CRPD that provide Long Term Security for Credential Verification Systems[9].

This paper proposes a comprehensive and decentralized Framework for the Authentication of Academic Certificates, whereby the Framework will provide Verifiable Evidence of both Validity through the combined verification of the Certificate and the issuing Institute. The Framework will use BlockChain Technology to provide Validation and Transparency of Academic Certificates and to eliminate any requirements for Centralized Authority. To maintain the Privacy of sensitive data, the Framework will be able to securely verify the validity of Academic Certificates using Zero Knowledge Proofs. Furthermore, to improve the speed at which Academic Certificates can be Verified and to reduce the computational load on the Verification Process, the Framework will employ a Hash-based Indexing Methodology for Certificate Lookup.

II. LITRATURE REVIEW

Akhmetshin et al., (2025) [26] In this paper, they used a lightweight hybrid blockchain solution for the verification of academic credentials, developed with Python and Docker. It incorporates the utilization of SHA-256 hashing, digital signatures, QR code technology, and BFT consensus. This proposed solution establishes a latency of 0.02 seconds, consuming fewer system resources. It succeeds Ethereum based systems with regards to efficiency, privacy, and modularity. Rehtaliani et al., (2025) [27] The authors proposed a decentralised approach for verifying credentials via Soulbound Tokens (SBTs), which are Ethereum EIP-4973 compliant. The credentials are tied to user wallets and are non-transferable, making them authentic. IPFS is employed for offline storage, and the process of verification and automatic revocation is performed through smart contracts. The proposed idea is conceptual and needs implementation in future research. Calderon et al., (2023) [28] This research paper discusses the utility of blockchain and NFT for the verification of academic certificates. Blockchain enables the assurance of immutability and transparency, whereas NFT enables the exclusive ownership of digital certificates. An inexpensive model is recommended for implementation in the educational sector. Obstacles for implementation are also addressed.Sharma et al., (2025) [29] The proposed study introduces a “privacy by design” framework for the academic verification process using the Oasis Sapphire confidentiality-preserving blockchain. The smart contracts are executed in a Trusted Execution Environment, and the data remains encrypted even after the verification process. The proposed system’s theoretical evaluation manifests cost efficacy and efficiency in the validation process.

Al Ahmed et al., (2024) [11] In the observation of the paper, CredChain is an Ethereum-based certificate verification platform for academics as well as professionals using IPFS. Smart contracts automate the process of certificate release, storage, and authentication. New algorithms increase authenticity. Experimentation reveals robust protection against certificate counterfeiting. S.H D and S. N., (2025) [12] This article proposes a new framework for the authentication of academic certificates based on the blockchain and IPFS using smart contracts from the Ethereum platform. This system allows certificates to be verified immediately and in a secure manner, and with no third-party involvement. The new system had very positive effects. Sultana et al., (2023) [13]The authors try to present a framework that uses IPFS, blockchain, and smart contracts to counter the problem of academic certificate fraud. The identity of the encrypted certificates is maintained in the blockchain, and the remaining data is in IPFS. Ethereum and MetaMask facilitate this process in a decentralized manner. Chen., (2025) [14] This work enhances the ECQV implicit certificate scheme by proposing the Certificate Digest Method. Only certificate digests are transmitted, and this lessens the packet size and computational complexity.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15553

Experiments on Raspberry Pi indicate efficiency gains. The scheme is appropriate for resource-constrained settings. Boudagdigue et al., (2023) [15] An overall trust-based certificate management framework for IIoT networks is proposed. Game theory and the update of beliefs regarding certificate revocation are used. The use of short-term certificates decreases overhead. Convergence to the optimal value is shown, and the amount of resources used is lowered. The time taken for revocation detection is

Li et al.,(2025) [23] Meta-BMEOC is a privacy-preserving blockchain-based mobile edge computing framework for the Metaverse. Smart contracts and threshold secret sharing are used for data privacy. TEEs provide accountability. The experimental results confirm the reduced latency and security improved. Moon., (2025) [24] The paper investigates blockchain knowledge and intentions for using blockchain services through surveys. The impact on intentions is strong for subjective knowledge and low for objective knowledge on intentions. Knowledge for adoption comes from education.

A. Contribution of the paper

Credential Verification and Issuer Authentication Integrated Approach: The proposed system allows for the verification of both the academic credentials and the issuing organisation in a decentralized construct, thereby creating greater confidence in the validity of these credentials compared to systems only verifying the credentials.

- Zero-Knowledge Proofs - Preserving Privacy in Credential Verification: Zero-knowledge proofs can be used to preserve the confidentiality of an individual's sensitive credential information when verifying the authenticity and correctness of the credentials.
- Mitigation of Replay Attacks with Self-Destroying Smart Contracts: Self-destructing smart contracts are introduced to prevent replay attack or the creation of counterfeit credentials.
- Post-Quantum Secure Digital Signatures: This framework uses lattice-based digital signatures –'s Private Key Cryptography– to provide a strong level of protection from potential quantum computing attacks.
- Fast Hash-Based Certificate Indexing: The proposed system uses a fast and efficient hashing mechanism to enable rapid retrieval of credential certificates and requires minimal on-chain storage.
- Blockchain Solutions are Cost Effective and Scalable: Testing of the proposed solution on a private Ethereum Blockchain shows low gas fees, fast credential verification, and an impressive overall credential verification rate of 94%; therefore, indicating that the proposed solution can be applied to large-scale or commercial applications.

III. METHODOLOGY: SYSTEM PRELIMINARIES, SYSTEM ARCHITECTURE AND IMPLEMENTATION OF PROPOSED WORK.

A. System Premilinaries

I. Cryptographic Hashing and Indexing

The cryptographic hash function (H(.) transforms the certificate data into a fixed Size of hash value (h). This hash acts as a unique fingerprint of the certificate. Any modification in the certificate data results in a different hash value due to the avalanche property of cryptographic hash functions. Therefore, the equation (h = H(certificate data))ensures data integrity, enables tamper detection, and allows efficient indexing and verification of certificates in decentralized systems.

$$h=H(certificatedata) \quad (1)$$

Where:

$H(.)$ = A cryptographic hash function
Certificate Data = The complete Data of the certificate (student name, course, issuer, date, etc.)
h = The resulting fixed-length hash value (hash digest).

II. Blockchain and Smart Contract Framework

The equation defines how each block in the blockchain is cryptographically linked to its predecessor. The hash of the previous block $Block_{i-1}$concatenated with the current block data $Data_i$, nd the combined input is passed through a cryptographic hash function H(.) to generate the hash of the current block $Block_i$. This creates a chain of blocks where each block depends on the hash of the previous one.

$$Block_i = H\,(Block_{i-1} \,||\, Data_i\,) \quad (2)$$

Where,

- $Block_i$ = Cryptographic hash of the current block (Block i).
- $Block_{i-1}$= Cryptographic hash of the previous block (Block i − 1).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

- $Data_i$ = Data stored in the current block, including:

  Certificate hash values
  Issuer identity proofs
  Smart contract execution results
  Timestamps and metadata
- H(.)= Secure cryptographic hash function.
- ||= Concatenation operation, meaning the previous block hash and current data are combined before hashing.

III. Privacy-Preserving Cryptography (ZKP + Lattice Signatures)

The Zero-Knowledge Proof mechanism allows a user to prove the validity of a certificate to a verifier without revealing the certificate's internal details such as student name, institution, or course information.The Verifier only obtains confirmation of the validity of the statement without obtaining any further information about the student Valid(Cert)=True. The result is a system that allows public verification while still maintaining privacy for all students using ZKP. Concurrently, Lattice-based digital signatures provide for the institutional signing of certificate hash values to ensure that only those institutions that are authorized to issue certificates will have the capability to sign them. Lattice-based signatures are also secure against attacks by quantum computers (unlike other older, more established, types of cryptography such as RSA and ECC) thereby making them a future-proof form of cryptography. Overall, ZKP provides protection against the unauthorized disclosure and/or misuse of data, while Lattice-based signatures provide for strong authentication of issuers, along with protections against replay or forgery attacks, creating an overall framework for an efficient and privacy-preserving verification process.

$$Valid(Cert)=True \qquad (3)$$

Where,

- Cert = Encrypted or hashed academic certificate data.
- Valid(Cert) = A Boolean verification function evaluated through a Zero-Knowledge Proof protocol.
- True = Indicates that the certificate satisfies all verification conditions (issued by authorized issuer, not revoked, not expired, and hash exists on blockchain).

B. *System Architecture*

The proposed architecture uses a decentralized and privacy-preserving structure to provide and confirm academic certification, using methods such as Blockchain Technology, IPFS, Bloom Filters, QR Codes and Zero Knowledge Proofs (ZKPs). This framework is guaranteed to provide certainty in regards to the validity of an academic certificate; furthermore, similar to the other frameworks listed above, it will provide a method of preventing certificate fraud and unauthorised certificate issuance while providing a scalable and private solution.

Authorized institutions are first registered through an Issuer Authorization Module and verified using a Know Your Customer (KYC) process. Upon successful verification, issuers are assigned a blockchain identity, establishing a trusted and accountable root of authority within the network. Certificate files are then uploaded by authorized issuers and stored off-chain using IPFS, which generates a unique Content Identifier (CID) for each certificate. Any modification to the file results in a different CID, enabling tamper detection.

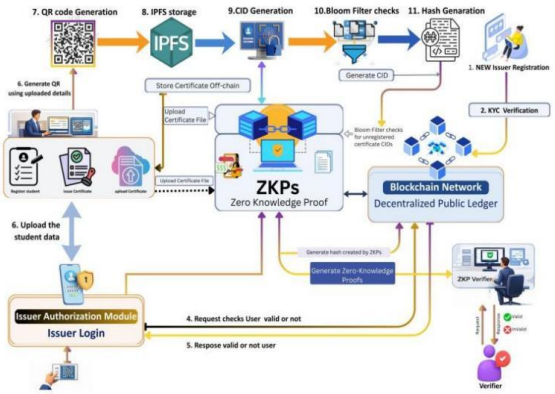

Fig. 1. System Architecture

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

C. Implementation Of Proposed Work

Step 1: New Issuer Registration

Let I denote the issuer identity data, which includes institutional credentials such as registration number, accreditation details, legal identity, and contact information. Let KYC(.) be the Know Your Customer (KYC) verification function that validates the legitimacy of an issuing institution. The authorization status of the issuer is computed as:

$$Auth_I = KYC(I) \quad (1)$$

Where $Auth_I \in \{True, False\}$.If $Auth_I$ =True, the issuer is considered verified and is authorized to participate in the system; otherwise, the registration request is rejected.

This step ensures that only legitimate and accredited institutions are permitted to issue certificates within the network. By enforcing issuer verification through a formal KYC process, the system prevents unauthorized or fraudulent entities from generating fake certificates. This establishes a trusted root of authority while maintaining decentralization, since only verified issuers can interact with the smart contracts responsible for certificate issuance.

- I =Issuer identity data (institution name, registration ID, accreditation, legal credentials).
- KYC(.)= Know Your Customer verification function that validates institutional legitimacy.
- $Auth_I$ = Authorization status of the issuer.
- True = Issuer is verified and allowed into the system.
- False = Issuer is rejected and not permitted to issue certificates.

1. Issuer Blockchain Identity Creation

Once an issuer is authorized, a cryptographic key pair PK_I , SK_I is generated to establish a secure digital identity. The public key is hashed to produce a unique blockchain identity ID_I, which is recorded on the blockchain and used to reference the issuer in all transactions. This process cryptographically binds the issuer to the blockchain, ensuring that every certificate issued can be traced back to a verified issuer and preventing identity spoofing or impersonation.

The issuer's key pair is generated as:

$$(PK_I, SK_I) = KeyGen(I) \quad (2)$$

The blockchain identity of the issuer is computed as:

$$ID_I = H(PK_I) \quad (3)$$

Where,

- I= Issuer identity data.
- KeyGen(.) = Cryptographic key generation function.
- SK_I = Issuer's private key used for signing certificates.
- PK_I= Issuer's public key used for signature verification.
- H(.) = Cryptographic hash function (e.g., SHA-256).
- ID_I = Issuer's blockchain identity.

Step 3: Student Data Upload

The student submits their information S to the system, which is securely stored using the Store(.) function. This step ensures that accurate and complete data is available for certificate generation and prevents errors during issuance.Sensitive fields may be encrypted or access-controlled to preserve privacy and prevent unauthorized access.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

$$\text{Store}(s) \qquad (4)$$

Where,

- S = Student data (name, registration number, course details, institution, issue date, etc.).
- Store(.) = Secure data storage function.

A. *step 4: Issuer Login Validation*

The issuer proves its identity by signing a challenge message using its private key. The system verifies this signature using the issuer's public key PK_I . If the verification function returns True, the issuer is authenticated and allowed to proceed with certificate issuance. This mechanism prevents unauthorized entities from accessing the system and ensures that only legitimate issuers can generate valid certificates.

$$\text{VerifyLogin}(PK_I , Sig_I) = \text{True} \qquad (5)$$

Where,

- PK_I = Issuer's public key.
- Sig_I = Digital signature generated using the issuer's private key SK_I.
- VerifyLogin(.) = Signature verification function.
- True = Issuer authentication is successful.

Step 5: Certificate Creation

The system constructs the certificate C by combining verified student data S, authenticated issuer identity I, course information, and issuance date. This ensures that each certificate is unique, traceable to a specific issuer, and correctly represents the awarded academic credential. The generated certificate is then ready for hashing, signing, and blockchain registration.

$$C = \text{Generate}(S, I, \text{Course}, \text{Date}) \qquad (6)$$

Where,

- C= Generated digital certificate file.
- S = Student data.
- I= Issuer identity.
- Course = Course or program name.
- Date = Certificate issuance date.
- Generate(.) = Certificate generation function.

Step 6: Upload Certificate to IPFS

The certificate Cis uploaded to IPFS, which returns a content identifier CID based on the cryptographic hash of the file. Since the identifier is content-addressed, any modification to the certificate will result in a different CID. This guarantees data integrity, tamper detection, and decentralized availability of certificate files without relying on centralized storage servers.

$$CID = IPFS(c) \qquad (7)$$

Where,

- C = Generated digital certificate file.
- IPFS(.) = InterPlanetary File System storage function.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

- CID = Content Identifier returned by IPFS.

Step 7: QR Code Generation

The content identifier CID uniquely represents the certificate stored on IPFS. By encoding CIDinto a QR code, the system enables verifiers to quickly retrieve and verify the certificate using a simple scan. This improves usability while maintaining cryptographic integrity, since the QR code contains only the content address and not the certificate data itself.

$$QR = Encode(CID) \quad (8)$$

Where,

- CID =IPFS content identifier of the certificate.
- Encode(.) = QR encoding function.
- QR = Generated QR code containing the encoded CID.

Step 8: Bloom Filter Check

The Bloom filter provides a fast and space-efficient method to check whether a certificate has already been issued or registered. If the Bloom filter indicates that the CID does not exist, the system inserts it and proceeds. If it already exists, the process is halted to prevent duplicate issuance or replay attacks. This mechanism protects the system from unauthorized or repeated certificate registrations while maintaining high performance.

The existence check is performed as:

$$\text{Exists} = BF(CID) \quad (9)$$

If the certificate does not exist, it is inserted into the Bloom filter:

$$BF(CID)=1 \quad \text{if Exists} = \text{False} \quad (10)$$

Where,

- BF = Bloom filter data structure.
- CID = IPFS content identifier of the certificate.
- Exists = Boolean result indicating whether the certificate is already registered.
- 1 = Bit insertion operation in the Bloom filter.

Step 9: Hash Generation

The content identifier CID is hashed to produce a fixed-length cryptographic fingerprint h_c. This fingerprint is stored on the blockchain instead of the full certificate, ensuring efficiency, immutability, and privacy. Any change to the certificate would change the CID, which in turn changes h_c, enabling reliable tamper detection.

$$h_c = H(CID) \quad (11)$$

Where,

- CID= IPFS content identifier of the certificate.
- H(.) = Cryptographic hash function (e.g., SHA-256).
- h_c = Hash of the certificate content identifier.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

Step 10: ZKP Proof Generation

The prover generates a proof π that it knows a value x whose hash equals the on-chain hash h_c, without revealing x itself. This allows a verifier to confirm the existence and authenticity of a certificate without learning any private information about the certificate or the student. This preserves privacy while still enabling public verifiability.To enable privacy-preserving verification, the system generates a zero-knowledge proof that the certificate exists and is valid without revealing the certificate itself.

The prover demonstrates knowledge of a value xxx such that:

$$\exists x : H(x)= h_c \qquad (12)$$

The corresponding zero-knowledge proof is generated as:

$$\pi=ZKP.Prove(x) \qquad (13)$$

Where,

- x = Secret witness (e.g., the certificate CID or preimage).
- H(.) = Cryptographic hash function.
- h_c = Hash stored on the blockchain.
- ZKP.Prove(.) = Zero-knowledge proof generation algorithm.
- π = Generated zero-knowledge proof.

Step 11: ZKP Verification

The verifier uses the verification function to check whether the proof π is valid with respect to the hash h_c. If the verification returns True, the verifier is assured that a valid certificate exists corresponding to the stored hash, without learning the certificate contents or the student's personal data. This ensures privacy-preserving public verification.

$$ZKP.Verify(\pi,h_c)=True \qquad (14)$$

Where,

- π= Zero-knowledge proof generated by the prover.
- h_c=Certificate hash stored on the blockchain.
- ZKP.Verify(.) = Zero-knowledge proof verification algorithm.
- True = Proof is valid and the certificate is authentic.

Step 12: Blockchain Storage

The transaction T_i contains the cryptographic reference to the certificate, the associated zero-knowledge proof, the issuer's public key, and a timestamp. This transaction is included in a block and linked to the previous block using a cryptographic hash, forming an immutable chain. Any modification to a previous block would change its hash and invalidate subsequent blocks, ensuring tamper resistance and permanent auditability.After successful proof verification, the certificate information is recorded on the blockchain as a transaction.

The transaction is defined as:

$$T_i= (h_c, \pi, PK_I, t) \qquad (15)$$

The transaction is then stored in a new block as:

$$B_i=H(B_{i-1}||T_i) \qquad (15)$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

Where,

- T_i= Blockchain transaction for certificate i.
- h_c= Certificate hash.
- π =Zero-knowledge proof.
- PK_I= Issuer's public key.
- t=Timestamp of issuance.
- B_i = Hash of the current block.
- B_{i-1} =Hash of the previous block.
- H(.) = Cryptographic hash function.
- || = Concatenation operation.

Step 13: Verification by Third Party

The verifier decodes the QR code to retrieve the certificate's content identifier and recomputes its hash. The Bloom filter confirms whether the certificate is registered in the system. The issuer's digital signature is verified to confirm that the certificate was issued by an authorized institution. Finally, the zero-knowledge proof is verified to ensure the certificate's authenticity without revealing any sensitive information. If all verification steps return True, the certificate is considered valid.When a third party scans the QR code on a certificate, the system performs a sequence of cryptographic checks to verify the authenticity and validity of the certificate.

First, the content identifier is extracted from the QR code:

$$CID'=Decode(QR) \quad (17)$$

The hash of the extracted identifier is computed as:

$$h'_c=H(CID) \quad (18)$$

The system then checks whether the certificate exists in the Bloom filter:

$$BF(CID') =True \quad (19)$$

Next, the issuer's signature is verified using the issuer's public key:

$$VerifyPK_i(h'_c,\sigma)=True \quad (20)$$

Finally, the zero-knowledge proof is verified:

$$ZKP.Verify(\pi,h'_c)=Tru \quad (21)$$

Where,

- QR= Scanned QR code from the certificate.
- Decode(.)=QR decoding function.
- CID' =Extracted IPFS content identifier.
- h'_c = Hash recomputed from the extracted CID.
- BF(.) = Bloom filter membership function.
- σ= Issuer's digital signature.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

- $VerifyPK_I(.)$ =Signature verification function using the issuer's public key.
- π = Stored zero-knowledge proof.
- ZKP.Verify(.)=Zero-knowledge proof verification algorithm.

IV. EXPERIMENTAL SETUP

The system initialization phase prepares the foundational components required for conducting the experimental evaluation. The blockchain ledger is initialized to ensure immutable storage of certificate references and issuer identities, while the Bloom filter is configured to support efficient certificate existence checks. IPFS storage is set up to enable scalable off-chain certificate storage, reducing blockchain overhead. Additionally, the Zero-Knowledge Proof framework is initialized to facilitate privacy-preserving verification throughout the experiment.

Algorithm 1A: Issuer Registration
Input: Issuer identity I

```
1: Auth_I ← KYC(I)
2: if Auth_I = False then
3:     Reject registration
4: else
5:     (SK_I, PK_I) ← KeyGen(I)
6:     ID_I ← Hash(PK_I)
7:     Store ID_I on Blockchain
8: end if
```

The issuer registration algorithm ensures that only legitimate institutions participate in the certificate issuance process. Each issuer undergoes a KYC-based identity verification, after which a cryptographic key pair is generated for secure signing operations. A unique issuer identifier is derived from the issuer's public key and permanently recorded on the blockchain. This process establishes a trusted authority layer and enforces accountability during experimental execution.

Algorithm 1B: Certificate Issuance
Input: Student data S, Issuer private key SK_I

```
1: C ← GenerateCertificate(S)
2: CID ← IPFS.Store(C)
3: if BF.Exists(CID) = True then
4:     Abort issuance (duplicate detected)
5: else
6:     BF.Insert(CID)
7: end if
8: h_C ← Hash(CID)
9: σ ← Sign(SK_I, h_C)
10: π ← ZKP.Prove(h_C)
11: T ← (h_C, σ, π, PK_I, timestamp)
12: Blockchain.Append(T)
13: QR ← EncodeQR(CID)
14: Attach QR to certificate
```

V. RESULT AND DISCUSSION

A. Performance metrics:

This section presents a detailed evaluation of the proposed decentralized certificate issuance and verification framework deployed on an Ethereum-compatible blockchain environment. The analysis primarily focuses on the cost efficiency, certificate search performance, comparison with existing blockchain-based approaches, comparison with NFT-based implementations using IPFS, and a comprehensive security assessment of the proposed architecture. In addition, a static security analysis of the deployed smart contracts is conducted to ensure robustness against potential vulnerabilities.

1) Cost analysis

This subsection analyzes the gas costs associated with executing the smart contract functions of the proposed system on the Ethereum blockchain. In Ethereum-based platforms, every transaction incurs a gas fee, which reflects the

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

computational effort required to execute the transaction and store data on the blockchain. The Remix IDE was utilized to estimate the gas consumption of smart contract deployment and function calls.

Gas costs are broadly classified into execution cost, which represents the computational expense of running smart contract logic, and transaction cost, which accounts for the cost of transmitting data to the blockchain network.

The total operational cost is calculated using

$$TC=\frac{EP\times TGC\times GP}{10^9} \qquad (22)$$

Let the total operational cost be defined as:

- TC denotes the total cost of operation in USD
- EP represents the Ether price in USD
- TGC is the total gas consumed
- GP is the gas price in gwei
- 1 Ether $=10^9$ gwei

TABLE I.

COST COMPARISON WITH EXISTING SOLUTIONS

Operation	Proposed System (USD)	Existing Blockchain (USD)	NFT-IPFS (USD)
Add Certificate	4.57	13.42	8.24
Add Institution	8.97	19.06	14.35

The proposed system significantly reduces operational cost by storing only cryptographic hashes and verification proofs on-chain. NFT-based systems incur additional minting and metadata costs, while traditional blockchain approaches suffer from high storage overhead.

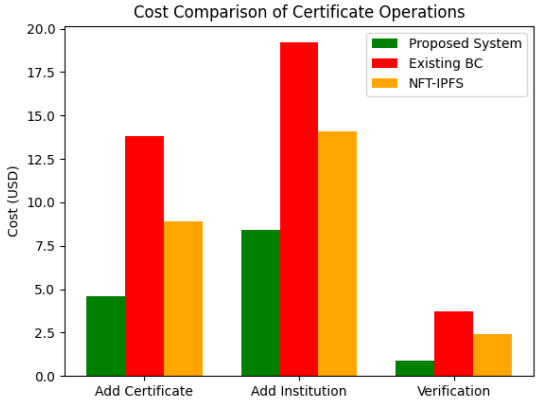

Fig. 2. Cost Comparison of Certificate operations

2) Certificate search time

Certificate search performance is evaluated under two scenarios:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

1. **False Case** – certificate does not exist
2. **True Case** – certificate exists

The search time is modeled as:

$$T_{search} = T_{BF} + T_{hash} + T_{verify} \quad (23)$$

where T_{BF} represents Bloom filter lookup time.

a) False Case Analysis

In the false case, the Bloom filter rapidly detects the absence of a certificate before querying the blockchain. This avoids unnecessary cryptographic verification.

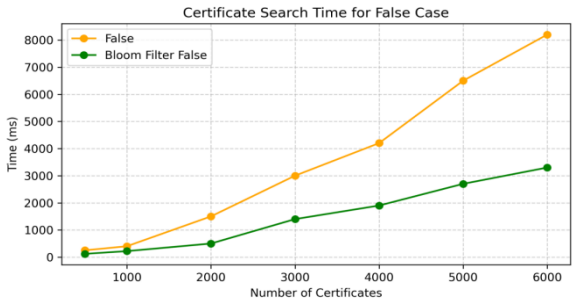

Fig. 3. Certificate search Time for False Case

b) True Case Analysis

In the true case, Bloom filter verification introduces a small overhead before blockchain validation. However, this overhead is minimal compared to the performance gain achieved in the false case.

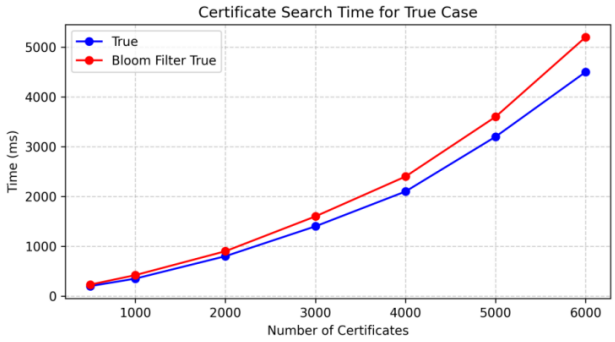

Fig. 4. Certificate search Time for True Case

3) Comparison of proposed system in terms of NFT using IPFS implementation

NFT-based certificate systems establish authenticity through ownership of digital tokens. However, academic certificates require verification of authenticity rather than ownership. In the proposed system, certificate data is hashed using SHA-256, and only the hash is stored on the blockchain.

Since SHA-256 produces a fixed-size output of 256 bits, the transaction size remains compact. Including blockchain metadata such as transaction identifiers, timestamps, and signatures, the total transaction size is approximately 182 bytes. As verification is performed using hashes, storing certificate files in IPFS is unnecessary, reducing system complexity and cost.

This approach ensures a lightweight, cost-effective, and privacy-preserving verification process without relying on NFT minting or token transfer mechanisms.

4) Security analysis of proposed architecture

a) 51% Attack

DOI: 10.17148/IJARCCE.2026.15553

The proposed system operates on Ethereum's Proof-of-Stake consensus, which significantly increases the economic cost of gaining majority control. Furthermore, certificate issuance is restricted to authorized institutions through a validation mechanism, making such attacks computationally and organizationally infeasible.

b) Sybil Attack

To mitigate Sybil attacks, each institution is restricted to a single blockchain identity. New issuers undergo verification and approval by existing trusted entities, preventing the creation of multiple fake identities.

c) Immutability

Once a certificate hash is recorded on the blockchain, it cannot be altered or deleted. Any modification to the certificate data results in a different hash, immediately invalidating forged certificates and ensuring long-term integrity.

d) Data Privacy

To preserve privacy, only cryptographic hashes and proofs are stored on-chain. The one-way nature of hash functions ensures that original certificate content cannot be reconstructed, protecting sensitive student information while maintaining verifiability.

5) Static security analysis of smart contract

Static security analysis was performed using the Slither framework, a widely used analysis tool for Ethereum smart contracts. Static analysis examines source code without execution, which is critical due to the immutable nature of deployed smart contracts.

6) Comparative Analysis

Authors & Year	Core Technology	Key Strength	Major Limitation	Privacy	Scalability	Cost	Issuer Authentication	Search Efficiency
Akhmetshin et al. (2025)	Hybrid Blockchain, SHA-256, BFT	Very low latency (0.02s)	No privacy or issuer validation	Low	Medium	Low	No	No
Rehtaliani et al. (2025)	SBTs, Ethereum, IPFS	Non-transferable credentials	Conceptual only	Medium	Medium	High	Partial	No
Calderón (2023	Blockchain, NFT	Simple ownership model	Privacy leakage, transferability	Low	Medium	Medium	No	No
Sharma et al. (2025)	Oasis Sapphire, TEE	Strong confidentiality	Expensive trusted hardware	High	Medium	High	No	No
Proposed Work	Blockchain + ZKP + Lattice Crypto + Bloom Filter	Privacy-preserving, quantum-safe, fast verification	Slight cryptographic complexity	High	High	Low	Yes	yes

VI. CONCLUSION

The proposed work combines the issuer verification and certificate verification methods into one system, creating a decentralized and protective way to verify academic credentials while also addressing the current issues associated with the creation and use of forged credentials and unauthorised credential issuers. This system prevents the use of forged credentials and unauthorised issuers by authenticating the issuer and validating the certificate on the same verification request.The proposed framework uses both cryptographic hash functions to provide unalterable certificates and zero-knowledge proof systems that allow verification without revealing sensitive personal information. This combination of technologies and techniques creates a robust system that protects against the future threats of quantum computing with the use of a lattice-based signature system.To maximise the speed and scalability of this system, both the verification latency and storage costs have been minimised by using a hashing-based indexing method with a Bloom filter. Using smart contracts to automate the verification process will ensure that the verification vendor does not have the ability to tamper with the system and that an attack of replay will not be possible.

DOI: 10.17148/IJARCCE.2026.15553

An experimental evaluation of this proposed framework on the Ethereum blockchain has shown that the average transaction cost for a verification is around 0.01 Ethereum, with the average times for retrieving the certificate being just under 2 seconds and the overall security efficiency of verification about 94%. These results demonstrate that this proposed framework is suitable for implementation in practice and can be a dependable and scalable method of guaranteeing the authenticity of academic credentials across many different types of higher education institutions.

REFERENCES

[1] R. -K. Kim, G. -S. Lee, J. -G. Park, H. Lee, S. -I. Moon and J. -W. Chang, "Optimal Scheduling and Commercial Testbed-Based Verification of Integrated PV-ESS Systems Considering Settlement Rules in South Korea," in IEEE Transactions on Sustainable Energy, vol. 16, no. 2, pp. 1372-1387, April 2025, doi: 10.1109/TSTE.2025.3529693.

[2] IEEE Draft Standard for Application Technical Specification of Blockchain-based E-Commerce Transaction Evidence Collecting," in IEEE P3802/D2.0, May 2021 , vol., no., pp.1-20, 22 Sept. 2021.

[3] Q. Zhang et al., "Exploring AIoT Blockchain Transaction Semantic Detection and Incentive Mechanism With Evolutionary Game Toward Web 3.0 Ecosystem," in IEEE Internet of Things Journal, vol. 12, no. 16, pp. 33243-33257, 15 Aug.15, 2025, doi: 10.1109/JIOT.2025.3575471.

[4] H. Huang et al., "BrokerChain: A Blockchain Sharding Protocol by Exploiting Broker Accounts," in IEEE Transactions on Networking, vol. 33, no. 4, pp. 1930-1945, Aug. 2025, doi: 10.1109/TON.2025.3550502.

[5] C. Fioravanti, C. N. Hadjicostis and G. Oliva, "A Control-Theoretical Zero-Knowledge Proof Scheme for Networked Control Systems," in IEEE Open Journal of Control Systems, vol. 3, pp. 416-428, 2024, doi: 10.1109/OJCSYS.2024.3455899.

[6] Wan, Y. Zhou and K. Ren, "zk-AuthFeed: Protecting Data Feed to Smart Contracts With Authenticated Zero Knowledge Proof," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1335-1347, 1 March-April 2023, doi: 10.1109/TDSC.2022.3153084.

[7] Pathak, I. Al-Anbagi and H. J. Hamilton, "Blockchain-Enhanced Zero Knowledge Proof-Based Privacy-Preserving Mutual Authentication for IoT Networks," in IEEE Access, vol. 12, pp. 118618-118636, 2024, doi: 10.1109/ACCESS.2024.3450313.

[8] Bradić, D. Delija, G. Sirovatka and M. Žagar, "Creating own NFT token using erc721 standard and solidity programming language," 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2022, pp. 1053-1056, doi: 10.23919/MIPRO55190.2022.9803593.

[9] Naidu, B. Wanjari, R. Bhojwani, S. Suchak, R. Baser and N. K. Ray, "Efficient Smart contract for Privacy Preserving Authentication in Blockchain using Zero Knowledge Proof," 2023 OITS International Conference on Information Technology (OCIT), Raipur, India, 2023, pp. 969-974, doi: 10.1109/OCIT59427.2023.10430710.

[10] M. Sifra, "Security Vulnerabilities and Countermeasures of Smart Contracts: A Survey," 2022 IEEE International Conference on Blockchain (Blockchain), Espoo, Finland, 2022, pp. 512-515, doi: 10.1109/Blockchain55522.2022.00080.

[11] Al Ahmed, R. A. Mamun Rudro, A. J. Prity, S. Saha, N. Mansoor and K. Nur, "CredChain: Academic and Professional Certificate Verification System using Blockchain," 2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS), Dhaka, Bangladesh, 2024, pp. 1-6, doi: 10.1109/iCACCESS61735.2024.10499520.

[12] S.H D and S. N, ""Secure Academic Certificate Authentication Using Blockchain Technology"," 2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2025, pp. 1-6, doi: 10.1109/CSITSS67709.2025.11294218.

[13] S. A. Sultana, C. Rupa, R. P. Malleswari, and T. R. Gadekallu, ''IPFS-blockchain smart contracts based conceptual framework to reduce certificate frauds in the academic field,'' Information, vol. 14, no. 8, p. 446, Aug. 2023

[14] A. C. H. Chen, "Comments on "L-ECQV: Lightweight ECQV Implicit Certificates for Authentication in the Internet of Things"—Efficiency Improvement Based on the Certificate Digest Method," in IEEE Access, vol. 13, pp. 93883-93891, 2025, doi: 10.1109/ACCESS.2025.3573266

[15] Boudagdigue, A. Benslimane, A. Kobbane and J. Liu, "Trust-Based Certificate Management for Industrial IoT Networks," in IEEE Internet of Things Journal, vol. 10, no. 14, pp. 12867-12885, 15 July15, 2023, doi: 10.1109/JIOT.2023.3259340.

[16] Tan-Vo et al., "Optimizing Academic Certificate Management With Blockchain and Machine Learning: A Novel Approach Using Optimistic Rollups and Fraud Detection," in IEEE Access, vol. 12, pp. 168135-168159, 2024, doi: 10.1109/ACCESS.2024.3486029.

[17] L. Sardar, "Fake Me If You Can: Unforgeable Digi-Physical Academic Certificates With Instant Verifiability," in IEEE Access, vol. 13, pp. 118334-118353, 2025, doi: 10.1109/ACCESS.2025.3583184.

[18] Rustemi, F. Dalipi, V. Atanasovski and A. Risteski, "A Systematic Literature Review on Blockchain-Based Systems for Academic Certificate Verification," in IEEE Access, vol. 11, pp. 64679-64696, 2023, doi: 10.1109/ACCESS.2023.3289598.

[19] M. Fartitchou, I. Lamaakal, K. E. Makkaoui, Z. E. Allali and Y. Maleh, "BlockMEDC: Blockchain Smart Contracts System for Securing Moroccan Higher Education Digital Certificates," in IEEE Access, vol. 13, pp. 39152-39175, 2025, doi: 10.1109/ACCESS.2025.3546177.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15553

[20] Ma, T. Feng, Q. Li and J. Xiong, "Blockchain-enabled Secure Distributed Data Aggregation and Verification Mechanism for IIoT," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 4383-4388, doi: 10.1109/GLOBECOM48099.2022.10000773.

[21] Ramzan, A. Aqdus, V. Ravi, D. Koundal, R. Amin and M. A. Al Ghamdi, "Healthcare Applications Using Blockchain Technology: Motivations and Challenges," in IEEE Transactions on Engineering Management, vol. 70, no. 8, pp. 2874-2890, Aug. 2023, doi: 10.1109/TEM.2022.3189734.

[22] Bellaj, A. Ouaddah, E. Bertin, N. Crespi and A. Mezrioui, "Drawing the Boundaries Between Blockchain and Blockchain-Like Systems: A Comprehensive Survey on Distributed Ledger Technologies," in Proceedings of the IEEE, vol. 112, no. 3, pp. 247-299, March 2024, doi: 10.1109/JPROC.2024.3386257.

[23] Li et al., "Blockchain-Based Privacy-Preserving and Accountable Mobile Edge Outsourcing Computing Framework for the Metaverse," in IEEE Transactions on Green Communications and Networking, vol. 9, no. 2, pp. 711-724, June 2025, doi: 10.1109/TGCN.2024.3451513.

[24] Moon, H. Chung, J. Ryu, K. Hwang, H. Moon and W. Park, "Blockchain Knowledge and Intention to Use Blockchain-Based Services," in IEEE Access, vol. 13, pp. 74521-74530, 2025, doi: 10.1109/ACCESS.2025.3562987.

