IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

BLOCKCHAIN-BASED CERTIFICATE VERIFICATION SYSTEM

[1]Aman Mathur

[1]Faculty of Computer Science,
[1]Lachoo Memorial College of Science and Technology

Abstract: Traditional certificate verification is slow, fraud-prone, and built on centralized systems that create obvious single points of failure. This paper proposes and implements a blockchain-based certificate verification system on Ethereum that addresses these problems at the architectural level. The system uses Solidity smart contracts for certificate lifecycle management, SHA-256 hashing for tamper detection, and IPFS for decentralized document storage. A role-based access control model, ECDSA digital signatures, and consensus-based validation protect against forgery and unauthorized changes. Four modules handle the core workflows: institutional issuance, student credential management, employer verification, and administrative governance. Testing on a private Ethereum network with 5,000 certificate transactions showed a 99.97% reduction in verification time compared to paper-based methods, zero successful fraud attempts, and improved system availability through distributed architecture.

Index Terms **- Blockchain, Certificate Verification, Smart Contracts, Ethereum, Distributed Ledger Technology, SHA-256 Hashing, Digital Credentials, IPFS, Solidity, Decentralized Applications.**

I. INTRODUCTION

Academic and professional credentials are how people prove what they know and what they have achieved. Employers, universities, and licensing bodies rely on them. So when the systems behind those credentials are unreliable, the downstream consequences are real: fraudulent hires, wasted institutional resources, and erosion of trust in credential-granting bodies.

The existing infrastructure has not kept up. Paper certificates can be forged with off-the-shelf software, and the barrier keeps dropping. A 2022 survey of 500 multinational corporations found that roughly 34% of applicants had misrepresented their academic qualifications to some degree [16] — a number that translates to billions of dollars in bad hires annually.

Digital verification systems helped somewhat, but centralization introduced a different set of problems. A single database is a single target. The 2021 ransomware attack on the Irish Health Service Executive showed what happens when centralized digital infrastructure goes down — and credential databases face exactly the same exposure [17].

Blockchain offers a different architectural assumption. Rather than trusting a central authority to maintain and protect the record, the record is distributed across a network where no single node can alter it unilaterally. Nakamoto's 2008 Bitcoin paper established the basic model [1]; Vitalik Buterin's 2014 Ethereum white paper extended it to support smart contracts [2] — programmable logic that runs on-chain without any intermediary.

This paper builds on that foundation. The proposed system lets institutions issue cryptographically signed certificates permanently recorded on Ethereum. Students receive a unique certificate hash. Employers verify authenticity in real time by querying the blockchain directly — no third-party service required, no email threads, no waiting days for a response.

The principal contributions of this research are:

- A decentralized certificate verification architecture built on Ethereum with IPFS integration for scalable document storage
- A smart contract suite implementing certificate issuance, verification, revocation, and transfer
- A multi-layer security framework using SHA-256 hashing, ECDSA digital signatures, and role-based access control
- Empirical evaluation across 5,000 test certificates demonstrating measurable improvements in speed, security, and cost
- A governance model for multi-institutional certificate issuance with cross-verification interoperability and audit trails

II. LITERATURE REVIEW

1 *Traditional Certificate Verification Systems*

Smith and Johnson [13] categorized certificate verification approaches into three generations: physical document examination, database-backed digital lookup, and cryptographic signature verification. Each addressed some failures of the previous generation while introducing new vulnerabilities. Physical inspection depended on trained examiners — a skill that forgery technology has largely outpaced. Database-backed systems sped things up but centralized the risk.

Thompson et al. [10] audited 120 UK university verification systems and found that 67% relied on email-based requests averaging 4.3 business days to complete. The same study found that 23% of institutions had no formal process for detecting fraudulent verification inquiries. The economic cost is substantial. According to Accredibase [14], over 3,000 degree mills operated globally, and the financial damage from credential fraud across G20 economies was estimated at USD 600 billion annually.

2 *Blockchain in Educational Credentialing*

Academic interest in blockchain-based credentialing picked up around 2016. Grech and Camilleri [3], writing for the European Commission's Joint Research Centre, identified credential verification as the highest-priority use case for blockchain in education. The core argument: blockchain's immutability addresses the root cause of certificate fraud rather than layering on detection mechanisms that can themselves be gamed.

MIT Media Lab's Blockcerts project [4][11] established an open standard for blockchain-based certificates using Bitcoin. It was a useful first step, but Bitcoin's limited programmability, variable transaction costs, and lack of native revocation mechanisms pushed subsequent research toward Ethereum. Chen et al. [5] proposed a distributed educational record system on Ethereum demonstrating sub-second verification times and 99.9% uptime.

Alammary et al. [6] addressed privacy gaps with a hybrid architecture combining a public Ethereum chain for certificate hash storage with a private consortium chain for sensitive student data. Zero-knowledge proofs enabled selective disclosure — graduates could prove specific attributes without revealing everything.

3 *Smart Contract Security*

Atzei et al. [7] catalogued twelve categories of Ethereum smart contract vulnerabilities. For credential management contracts, improper access control is the critical one — an unauthorized issuance completely undermines the system. Wang and Su [8] analyzed five blockchain-based certificate systems and identified three common weaknesses: inadequate role-based access control, no revocation mechanism, and missing event logging for audit trails. Their remediation framework — using OpenZeppelin access control, Merkle proof-based revocation lists, and comprehensive event emission — forms the basis for the security design in this paper.

4 *Research Gaps*

Most proposed systems test on small datasets (under 1,000 certificates), leaving scalability at production loads unexamined. User experience for non-technical stakeholders has received little attention. Integration with legacy institutional systems is underaddressed. Governance models for multi-institutional consortia, including dispute resolution and certificate transfer protocols, are largely underdeveloped. This paper addresses each through system design, large-scale experimental evaluation, and stakeholder-centered interface design.

III. System Architecture and Design

5 *Architectural Overview*

The system uses a three-tier architecture: a presentation layer, an application logic layer, and a data persistence layer. Unlike conventional three-tier designs where the application and data layers sit on centralized servers, both layers here are distributed across Ethereum and IPFS. This eliminates single points of failure, ensures data permanence, and removes the need to trust any single organization's infrastructure.

The presentation layer consists of web and mobile interfaces for three user roles: institutional administrators handling certificate issuance, students managing and sharing credentials, and verifiers (employers, academic institutions, licensing bodies) checking authenticity.

6 *System Components*

The system has seven interconnected components:

- Institution Module: Web portal for authorized administrators to upload certificate data, trigger issuance transactions, and manage access credentials. Batch issuance handles up to 10,000 certificates per transaction batch.
- Student Module: Progressive web application for graduates to view, download, and share blockchain-verified credentials via unique certificate links or QR codes, with direct integration to LinkedIn and major job platforms.
- Verification Module: Publicly accessible portal where employers enter a certificate ID or scan a QR code for real-time authenticity confirmation. Supports API integration for automated background check workflows.
- Smart Contract Layer: Solidity contracts on Ethereum implementing issuance, verification, revocation, and transfer logic. Upgradeable via the OpenZeppelin Transparent Proxy pattern.
- IPFS Storage Layer: Distributed file storage for certificate documents. Only IPFS content hashes are stored on-chain, keeping transaction costs down while maintaining tamper-proof document integrity.
- Oracle Service: Chainlink oracle integration giving smart contracts access to off-chain data — such as institutional accreditation status — for automated validation of issuing authority legitimacy.
- Administrative Governance Module: Multi-signature interface for consortium members to propose, vote on, and execute system-level changes, including adding new institutional issuers.

7 *Certificate Data Model*

Each certificate has a structured data model split between on-chain metadata and off-chain document content. The on-chain record stores the minimum needed for verification; the full document lives on IPFS.

Field	Data Type	Storage	Description
certificateId	bytes32	On-chain	SHA-256 hash of certificate content
studentName	string	On-chain	Full name of certificate holder
enrollmentNo	string	On-chain	Unique student enrollment number
degreeName	string	On-chain	Academic degree or certification title
issuerAddress	address	On-chain	Ethereum address of issuing institution
issueTimestamp	uint256	On-chain	Unix timestamp of issuance
expiryDate	uint256	On-chain	Expiry timestamp (0 = non-expiring)
ipfsHash	string	On-chain	IPFS CID of full certificate document
isRevoked	bool	On-chain	Revocation status flag
digitalSignature	bytes	On-chain	ECDSA signature of certificate hash
certificateDocument	PDF/Image	IPFS	Full certificate document file
supportingDocuments	Array	IPFS	Marksheets and supporting evidence

Table 1: Certificate Data Model Fields and Storage Locations

8 *Workflow Design*

Certificate Issuance Workflow

- Step 1: The Administrator uploads the student's data together with the certificate file using the Institution Module.
- Step 2: The system uploads the certificate file to IPFS and gets its content hash (CID).
- Step 3: The backend generates the SHA-256 hash based on the certificate data.

- Step 4: The Administrator signs the hash using the institution's private key creating an ECDSA signature.
- Step 5: Smart contract issueCertificate() is called with certificate data, signature, and IPFS CID.
- Step 6: The smart contract checks that the caller is an authorized institution, registers the record, and generates the CertificateIssued event.
- Step 7: The transaction is broadcasted to the Ethereum network, verified, and registered.
- Step 8: The student gets the certificate ID (hash) bytes32 and QR code.

Certificate Verification Workflow

- Step 1: The verifier types the certificate ID into the module or scans the QR code using the Verification Module.
- Step 2: The frontend requests smart contract the verifyCertificate() function passing the certificate ID.
- Step 3: Contract retrieves the certificate record and checks revocation status.
- Step 4: System validates the digital signature against the institution's registered public key.
- Step 5: Verifier receives complete certificate details with authenticity confirmation or fraud alert within 3 seconds.

IV. SMART CONTRACT IMPLEMENTATION

[9] *Contract Architecture*

The contract suite has three primary contracts: CertificateRegistry.sol, InstitutionManager.sol, and GovernanceController.sol. They interact through defined interfaces, enabling modular upgradability and independent security auditing. All contracts inherit from OpenZeppelin's base implementations for access control, pausability, and reentrancy guard [15].

[10] *Core Smart Contract Functions*

Certificate Issuance Function

```
function issueCertificate(
    bytes32 _certificateId, string memory _studentName,
    string memory _enrollmentNo, string memory _degreeName,
    uint256 _expiryDate, string memory _ipfsHash,
    bytes memory _signature
) external onlyInstitution nonReentrant {
    require(!certificates[_certificateId].exists, "Already issued");
    require(_verify(_certificateId, _signature), "Invalid sig");
    certificates[_certificateId] = Certificate({
        studentName: _studentName, enrollmentNo: _enrollmentNo,
        degreeName: _degreeName, issuer: msg.sender,
        timestamp: block.timestamp, expiry: _expiryDate,
        ipfsHash: _ipfsHash, isRevoked: false, exists: true
    });
    emit CertificateIssued(_certificateId, msg.sender, _studentName);
}
```

Certificate Verification Function

```
function verifyCertificate(bytes32 _certId)
    external view returns (bool valid, string memory status, Certificate memory cert) {
    cert = certificates[_certId];
    if (!cert.exists) return (false, "NOT_FOUND", cert);
    if (cert.isRevoked) return (false, "REVOKED", cert);
    if (cert.expiry != 0 && block.timestamp > cert.expiry)
        return (false, "EXPIRED", cert);
    return (true, "VALID", cert);
}
```

Certificate Revocation Function

```
function revokeCertificate(bytes32 _certId, string memory _reason)
    external onlyInstitution {
    require(certificates[_certId].exists, "Not found");
    require(certificates[_certId].issuer == msg.sender, "Unauthorized");
    certificates[_certId].isRevoked = true;
    emit CertificateRevoked(_certId, msg.sender, _reason);
}
```

11 *Security Architecture*

1) Cryptographic Hash Integrity

Every certificate is identified by its SHA-256 hash, computed over all certificate fields including student information, institutional metadata, and issue timestamp. The avalanche property of SHA-256 means any change to the certificate content — even a single character — produces a completely different hash, making tampering immediately detectable at verification. The hash serves as the certificate's unique identifier, its integrity proof, and its blockchain storage key, giving O(1) lookup time regardless of how many certificates are in the system.

2) ECDSA Digital Signature Scheme

Institutional authenticity is enforced through ECDSA signatures using the secp256k1 curve — the same cryptographic primitive Ethereum uses for transaction signing [12]. Each participating institution registers its public key in the InstitutionManager contract. Certificate issuance requires signing the certificate hash with the institution's private key before calling the smart contract. The contract's internal _verify() function uses Solidity's built-in ecrecover() precompile to recover the signer's address from the signature and compare it against the registered institution address.

3) Role-Based Access Control

The system uses a four-level RBAC model: ADMIN_ROLE for system-level governance, INSTITUTION_ROLE for certificate issuance and revocation, AUDITOR_ROLE for read-only access to full audit logs, and the default public role for verification queries. Role assignments are managed through OpenZeppelin's AccessControl contract. ADMIN_ROLE operations require multi-signature approval to prevent unilateral system changes.

V. EXPERIMENTAL EVALUATION

12 *Experimental Setup and Environment*

Experiments ran across two environments: a private Ethereum test network simulating mainnet conditions, and the public Ethereum Sepolia testnet for cross-network validation. The private network had twelve nodes across three geographic regions (North India, South India, West India) to simulate realistic latency. Each node ran Geth v1.11.0 with Clique proof-of-authority consensus and a 3-second block time. The test dataset contained 5,000 synthetic certificates. All experiments ran five independent trial runs and results were averaged.

Parameter	Configuration
Blockchain Platform	Ethereum (Geth v1.11.0)
Consensus Mechanism	Clique Proof-of-Authority
Network Nodes	12 nodes across 3 regions
Block Time	3 seconds
Smart Contract Language	Solidity v0.8.19
Development Framework	Hardhat v2.14.0
Frontend Framework	React.js v18.2.0
Blockchain Interaction	ethers.js v6.3.0
IPFS Implementation	Kubo v0.19.1
Test Dataset Size	5,000 certificates
Trial Runs	5 independent runs

Table 2: Experimental Environment Configuration

13 *Performance Results*

1) Verification Time Analysis

System Type	Min Time	Max Time	Mean Time	Std Dev
Paper-Based (manual)	3 days	7 days	4.8 days	1.2 days
Email Verification	1 hour	5 days	18.3 hours	6.7 hours
Centralized Digital DB	45 sec	300 sec	127 sec	48 sec
Proposed Blockchain System	0.8 sec	4.2 sec	2.1 sec	0.6 sec

Table 3: Verification Time Comparison Across Systems

The blockchain system averaged 2.1 seconds per verification — 99.97% faster than paper-based methods averaging 4.8 days. Against the best-performing centralized digital alternative (mean 127 seconds), the blockchain system is 60x faster. This comes from the blockchain's indexed key-value storage, which gives O(1) certificate lookups regardless of total database size.

2) Transaction Throughput and Scalability

Under simulated peak load with 500 simultaneous issuance requests, the system sustained 47 transactions per block (roughly 15.7 certificates per second on a 3-second block time). Batch processing pushed effective throughput to 312 certificates per second for bulk operations — enough for semester-end mass issuance at large universities.

Load Level	Concurrent Users	Throughput (cert/sec)	Avg Latency	Error Rate
Light	10	15.7	1.8 sec	0.00%
Moderate	100	15.6	2.1 sec	0.00%
Heavy	500	15.4	2.8 sec	0.02%
Stress	1000	14.9	4.1 sec	0.08%
Batch Mode	N/A	312.0	0.9 sec	0.00%

Table 4: Scalability Metrics Under Different Load Conditions

3) Gas Cost Analysis

Certificate issuance consumed an average of 187,432 gas units — roughly USD 0.23 at May 2024 mainnet prices (25 gwei, ETH at USD 3,000). Verification is a read-only call and consumes no gas. Revocation averaged 42,891 gas (approximately USD 0.05). For institutional-scale deployment, Layer-2 deployment on Polygon or Arbitrum reduces costs by an estimated 95%. The architecture accommodates this without changes to the core smart contract logic.

14 ***Security Analysis***

Attack Vector	Traditional System	Centralized Digital	Proposed Blockchain
Certificate Forgery	Vulnerable	Moderate Risk	Immune (hash-based)
Data Tampering	Vulnerable	Moderate Risk	Immune (immutable)
Single Point Failure	High Risk	High Risk	Eliminated
Replay Attack	Vulnerable	Partial Defense	Protected (nonce)
Sybil Attack	N/A	N/A	Protected (RBAC)
DDoS Attack	High Impact	High Impact	Low Impact
Insider Threat	High Risk	High Risk	Auditable (on-chain)
Man-in-the-Middle	Vulnerable	TLS Protected	Cryptographically Secure

Table 5: Security Assessment Against Common Attack Vectors

The proposed system satisfies four fundamental security properties. Completeness: every valid certificate can be successfully verified — all 5,000 test certificates were retrieved correctly. Soundness: penetration testing with 200 crafted fraudulent certificate hashes resulted in 100% rejection. Non-repudiation: the immutable record and ECDSA signature provide cryptographic proof that the issuing institution cannot deny. Forward Secrecy: past certificates remain verifiable even if an institution's current private key is later compromised.

VI. DISCUSSION

15 ***Comparison with Existing Systems***

Compared to Blockcerts, the Ethereum-based system provides richer programmability through smart contracts, supporting automated revocation, expiry management, and multi-institutional governance that Bitcoin-based implementations cannot easily accommodate. IPFS integration resolves the scalability problem of storing full certificate data on-chain while preserving tamper-proof integrity through content-addressed storage.

Against centralized digital verification services, the proposed system removes the commercial dependency and annual subscription costs. Organizations can achieve equivalent functionality through one-time smart contract deployment, with ongoing costs limited to transaction fees — which approach zero in consortium blockchain deployments.

16 ***Limitations***

Ethereum mainnet gas prices vary with network congestion. During high-activity periods, individual certificate issuance can become expensive. Layer-2 migration addresses this but requires additional engineering and introduces bridge security considerations. Private key management is a real operational challenge — if an institution's key is compromised, unauthorized certificates can be issued until revocation is processed through the governance contract. Hardware security module (HSM) integration is recommended for production deployments. Graduates unfamiliar with blockchain concepts may find the credential sharing workflow unintuitive, requiring institutional onboarding support.

VII. CONCLUSION

This paper presented a blockchain-based certificate verification system designed to address the security, efficiency, and trust failures of traditional credential verification. Built on Ethereum smart contracts with IPFS storage and ECDSA cryptographic signatures, the system provides security guarantees that centralized architectures cannot match.

Testing across 5,000 certificates on a multi-node private Ethereum network showed a 99.97% reduction in verification time compared to paper-based methods, 100% fraud detection against crafted fake certificates, and throughput sufficient for mass institutional deployment. The hierarchical RBAC model, multi-signature governance, and comprehensive audit logging provide the governance structure needed for responsible multi-institutional use.

The design philosophy — minimal on-chain data storage paired with IPFS for documents — balances blockchain security properties against operational cost. The modular smart contract architecture supports future upgrades without data migration. The primary barriers to widespread adoption are not technical but organizational: institutions need to participate in shared consortia and invest in cryptographic key infrastructure. Overcoming these organizational barriers is the most impactful direction for future work.

VIII. FUTURE WORK

- Zero-Knowledge Proof Integration: Implementing zk-SNARKs to enable selective credential disclosure — proving graduation status or degree classification without revealing full certificate details.
- Cross-Chain Interoperability: Bridge protocols enabling certificates issued on one blockchain to be verifiable on alternative networks.
- AI Integration: Machine learning models for anomaly detection in certificate issuance patterns, flagging potential fraudulent issuance by compromised institutional accounts.
- Mobile-First Architecture: Native mobile applications with offline verification using locally cached certificate hashes.
- ISO/IEC Standards Alignment: Mapping the data model to ISO/IEC 21000, W3C Verifiable Credentials, and Open Badges for global interoperability.
- Decentralized Identifier (DID) Integration: Replacing Ethereum address-based institution identification with W3C Decentralized Identifiers.
- Quantum-Resistant Cryptography: Evaluating post-quantum signature schemes such as CRYSTALS-Dilithium as replacements for ECDSA.

IX. REFERENCES

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” *Cryptography Mailing List*, Oct. 2008.

[2] V. Buterin, “Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform,” 2014.

[3] A. Grech and A. F. Camilleri, *Blockchain in Education*, European Commission Joint Research Centre, EUR 28778 EN, 2017.

[4] M. Sharples and J. Domingue, “The Blockchain and Kudos: A Distributed System for Educational Record, Reputation and Reward,” in *Adaptive and Adaptable Learning: Proceedings of EC-TEL 2016*, Lecture Notes in Computer Science, vol. 9891, Springer, Cham, Switzerland, 2016, pp. 490–496, doi: 10.1007/978-3-319-45153-4_48.

[5] G. Chen, B. Xu, M. Lu and N.-S. Chen, “Exploring blockchain technology and its potential applications for education,” *Smart Learning Environments*, vol. 5, no. 1, pp. 1–10, 2018, doi: 10.1186/s40561-017-0050-x.

[6] A. Alammary, S. Alhazmi, M. Almasri and S. Gillani, “Blockchain-Based Applications in Education: A Systematic Review,” *Applied Sciences*, vol. 9, no. 12, p. 2400, 2019, doi: 10.3390/app9122400.

[7] N. Atzei, M. Bartoletti and T. Cimoli, “A Survey of Attacks on Ethereum Smart Contracts,” in *Proceedings of POST 2017*, 2017, pp. 164–186, doi: 10.1007/978-3-662-54455-6_8.

[8] X. Wang and Y. Su, “Blockchain-Based Educational Credential Verification: Security Analysis and Framework Design,” *IEEE Access*, vol. 8, pp. 225606–225619, 2020, doi: 10.1109/ACCESS.2020.3046234.

[9] L. Luu *et al*., “Making Smart Contracts Smarter,” in *Proceedings of the ACM CCS 2016*, 2016, pp. 254–269, doi: 10.1145/2976749.2978309.

[10] R. Thompson, A. Clark and P. Williams, “Audit of UK University Certificate Verification Systems,” *Journal of Higher Education Policy and Management*, vol. 39, no. 4, pp. 412–428, 2017, doi: 10.1080/1360080X.2017.1330802.

[11] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps*. Sebastopol, CA, USA: O’Reilly Media, 2018.

[12] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” *Ethereum Project Yellow Paper*, 2014.

[13] J. Smith and M. Johnson, "A Taxonomy of Academic Credential Verification Systems: Three Generations of Practice and Vulnerability," Journal of Educational Technology and Society, vol. 18, no. 3, pp. 142–157, 2015.

[14] Accredibase, "The Global Degree Mill Industry: Scale, Impact, and Economic Cost," Accredibase Annual Report, 2020. [Online]. Available: https://accredibase.com/reports/2020

[15] OpenZeppelin, "OpenZeppelin Contracts: Battle-tested smart contract library," GitHub Repository, 2023. [Online]. Available: https://github.com/OpenZeppelin/openzeppelin-contracts

[16] HireRight, "2022 Global Benchmark Report: Employment Screening Trends," HireRight, 2022. [Online]. Available: https://www.hireright.com/resources/benchmarking-reports

[17] Health Service Executive, "Conti Cyberattack on the HSE Ireland: A Report on the Ransomware Attack," HSE Ireland, Dec. 2021. [Online]. Available: https://www.hse.ie/eng/services/news/media/pressrel/hse-publishes-report-on-the-conti-cyberattack.html