International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

Certificate Validation Using Blockchain

Mayuri Nagare[1], Aadi Mahalle[2], Digvijay Dakhore[3], Mrs. Soudamini Somvanshi[4], Mrs. Supriya Sathe[5]
[1,2,3]UG - Computer Engineering, Pune, 411044, India
[4,5]Assistant Professor, Computer Engineering, Pune, 411044, India
Emails id: *mayurinagare234@gmail.com[1], aadimahalle5@gmail.com[2], yashdakhore15@gmail.com[3], 4stsomvanshi@dypcoeakurdi.ac.in[4], sbsathe@dypcoeakurdi.ac.in[5]*

Abstract
Conventional certificate verification, whether paper-based or backed by a centralized digital registry, remains exposed to forgery, single points of failure, and slow manual cross-checking between issuers, holders, and verifiers. This paper presents a decentralized, blockchain-based framework for issuing, storing, and verifying academic and professional certificates that addresses these weaknesses without placing the full document on-chain. Each certificate is reduced to a SHA-256 hash, signed with the issuing institution's RSA private key, and recorded through an Ethereum smart contract, while the original file is retained off-chain on IPFS and referenced by its Content Identifier. Only institutions that pass a unanimous, vote-based onboarding process administered by a validator consortium are permitted to issue certificates, which constrains the system to a trusted-issuer model while preserving decentralization across the validator set. A hash-mapped Bloom Filter sits in front of the blockchain query path and performs a fast probabilistic existence check, allowing forged or non-existent certificates to be rejected before an on-chain lookup is triggered. A prototype was implemented with Solidity smart contracts on the Ethereum Sepolia testnet, a Node.js/Web3.js application layer, and a React.js frontend with MetaMask-based authentication. Evaluation on the testnet shows that the Bloom Filter pre-check lowers the average lookup time for invalid certificates by roughly 87%, keeps verification of valid certificates under two seconds end-to-end, holds the false-positive rate below 0.5% for up to 50,000 stored certificate hashes, and reduces the gas cost of issuing a certificate to approximately 4.57 USD. These results indicate that combining consortium governance, cryptographic hashing, and probabilistic filtering yields a certificate validation pipeline that is simultaneously tamper-resistant, low-cost, and fast enough for real-time institutional use.

Keywords: *Blockchain, Certificate Verification, Smart Contracts, Ethereum, Bloom Filter, IPFS, RSA Digital Signature, Issuer Validation.*

1. Introduction

Academic and professional certificates are widely used to verify an individual's qualifications and achievements. However, traditional certificate verification systems are vulnerable to forgery, unauthorized modifications, and delays caused by manual verification processes. Centralized digital storage systems improve accessibility but still suffer from issues such as single points of failure, data breaches, and limited transparency. Blockchain technology provides a secure and decentralized approach to storing and verifying digital records. Since blockchain records are immutable and distributed across multiple nodes, unauthorized modifications become extremely difficult. These properties make blockchain a suitable platform for certificate management and verification. Despite the advantages of blockchain-based systems, several challenges remain. Storing complete certificate files on-chain increases storage costs, while many existing solutions do not verify the authenticity of issuing institutions. Additionally, verifying invalid certificates may still require unnecessary blockchain

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

queries, leading to increased response time and resource consumption. To address these challenges, this paper proposes a blockchain-based certificate validation framework that combines SHA-256 hashing, RSA digital signatures, IPFS storage, consortium-based issuer validation, and Bloom Filter-based pre-verification. The proposed approach improves security, reduces verification time, and minimizes storage and transaction costs while maintaining transparency and trust. The remainder of this paper is organized as follows. Section II presents the literature review. Section III describes the system architecture and methodology. Section IV explains the core algorithms. Section V discusses implementation and results. Section VI presents the security analysis, and Section VII concludes the paper with future research directions.

2. Literature Review

A system named EduCTX emerged through work by Turkanović and colleagues, built on blockchain technology to support movement of academic credits across universities [2]. While proof came forward that shared digital ledgers can manage records among schools, focus stayed fixed on point exchange instead of confirming single credential files. This emphasis opened space for new solutions targeting validation at the level of diplomas or degrees. The original framework did not close the need for tools verifying official graduation documents. Ghani et al. applied a controlled-access blockchain, built on Hyperledger Fabric, to manage academic credentials [6]. Focused on traceability and restricted entry, their model operated within one institution only. Yet high initial investment and intricate maintenance make such networks poorly aligned with environments involving multiple institutions where minimal setup effort is essential. A different path becomes necessary when inclusion and simplicity are central requirements. LightLedger, presented by Garba et al., combined Bloom Filters and blockchain storage to speed up certificate searches, while applying Zero-Knowledge Proofs to preserve confidentiality [3]. Though its outcomes showed that filtering through Bloom structures notably cuts response time when checking unverified certificates - aligning closely with findings detailed in Section V here - the design relied on browser extensions for user access. Instead of choosing validators at random, the approach now uses a fixed group selected via voting, improving consistency. Client engagement occurs through common MetaMask wallets rather than specialized add-ons, broadening reach without extra setup. Such changes sidestep limitations tied to plugin dependency and unpredictable node assignment seen earlier. A study by Rahman et al. tackled one issue in blockchain-based credentials: how fixed records clash with real-world corrections [4]. Instead of removing data, their method allows revisions using two linked chains. This structure enables modifications under defined conditions. Yet it does not rely on agreement across multiple institutions. As a result, deployment[9] in networks with diverse issuing bodies becomes difficult. In contrast, the technique described in Section IV maintains traceable changes via cancellation and re-creation. It operates within a shared, single ledger managed collectively. History remains intact even when adjustments occur. Design choices differ, though objectives show overlap. Beginning elsewhere than education, Pericás-Gornals and team used blockchain together with proxy re-encryption for digital proof of COVID-19 status, guarding personal health details without blocking verification [1]. Although this method works well for medical records, it does not extend easily to degrees or job qualifications; meanwhile, Adja's group introduced a public key system powered by blockchain focused on cancelling certificates more efficiently - yet demands greater agreement effort among nodes, slowing response times [7]. Beginning near current efforts, one study introduced a method using Bloom Filters alongside RSA signatures and a vote-driven approval process for issuing bodies, noting reduced transaction fees and fast searches despite some setup delay when adding participants [10]. Rather than focusing on adaptive verification, another team proposed a permanent, worldwide format for digital credentials yet omitted mechanisms that adjust trust as new issuers join [9]. What follows takes elements - issuer checks and filter structures - from both earlier works [10] and [5], though it shifts data hosting off-chain via IPFS and introduces a clear way to fix certificates if needed; instead of relying only on models, performance is checked through real-world gas usage and response time tests on an active

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

Ethereum test environment,as shown in Table 1 Summary of Related Blockchain Certificate Verification Approaches[8].

Table 1 Summary of Related Blockchain Certificate Verification Approaches

Study	Technique	Limitation Addressed
EduCTX [2]	Credit-transfer ledger	No certificate-level check
Ghani et al. [6]	Hyperledger Fabric	High setup cost
LightLedger [3]	Bloom Filter + ZKP	Plugin dependency
Rahman et al. [4]	Dual-chain correction	No multi-party consensus
Priyadarshini et al. [10]	Bloom Filter+RSA+voting	Onboarding cost

3. System Architecture and Methodology

The structure includes four levels: Frontend, Application, Blockchain, followed by Storage. Each handles separate stages[11] of certificates, beginning with creation. Verification occurs later in the sequence. Corrections happen only if required. Responsibility shifts across layers without overlap.

3.1.Layered Architecture

What lies behind the screen is a React.js interface, one shaped differently depending on whether the user represents an institution, a student, or a verifier. Connected through this view, MetaMask steps in - not visible but active - managing digital identity via cryptocurrency wallets when actions require verification. Further inside runs a coordination tier based on Node.js, supported by tools that link web processes directly to decentralized networks. From there, operations unfold quietly: data receives cryptographic fingerprints[12] using SHA-256, digital seals form and are examined under RSA rules, interactions get routed across system boundaries without overlap. Beneath those moving parts rests code written in Solidity, awake on the Ethereum Sepolia network. Its purpose remains narrow - registering proof of certificates, overseeing approval for new issuers, limiting function access strictly to authorized roles. Separate yet linked, another component holds full documents outside the chain altogether. That place is IPFS, where files live independently; their presence confirmed only by unique labels stored within smart contracts - compact references replacing bulk, preserving efficiency without sacrificing traceability[13].

3.2.Cryptographic and Probabilistic Models

From each certificate file (c), a SHA-256 hash generates a fixed-size output represented as (H(c)). Let (CH) denote the numerical representation of (H(c)). The position (FN) in the Bloom Filter is calculated as:

$$FN = CH \bmod TI \tag{1}$$

where (TI) represents the total number of bits available in the Bloom Filter. This mechanism enables rapid verification without inspecting the complete certificate file.

Using an RSA private key (d) and modulus (n), an issuing organization signs the certificate hash by computing:

$$S = M^d \bmod n \tag{2}$$

where (M) is the hash-derived message. Verification is performed using the issuer's public exponent (e):

$$M' = S^e \bmod n \tag{3}$$

The certificate is considered authentic only when (M = M'). This process ensures both data integrity and issuer authenticity through RSA-based digital signatures.

Bloom Filter sizing depends on the expected number of certificate records. Let (n) be the number of stored certificates, (p) the acceptable false-positive probability, (m) the filter size, and (k) the number of hash functions. The optimal filter size is calculated as:

$$m = -\frac{(n \ln(p))}{(\ln 2)^2} \tag{4}$$

The optimal number of hash functions is:

$$k = \left(\frac{m}{n}\right) \ln 2 \tag{5}$$

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

The false-positive probability is estimated by:

$$p \approx \left(1 - e^{-\frac{kn}{m}}\right)^k \quad (6)$$

These equations were used to configure the Bloom Filter parameters in the proposed system, ensuring efficient performance while maintaining a low false-positive rate[14]. The cost of Ethereum transactions is computed using the gas consumed by an operation, the current gas price (in gwei), and the market value of Ether (ETH). The transaction cost is given by:

$$Transaction\ Cost\ (USD) = Gas\ Used \times Gas\ Price \times \frac{ETH\ Price}{10^9} \quad (7)$$

This formula converts the transaction cost from gwei to USD and is used to evaluate the economic feasibility of certificate issuance and institution onboarding operations.

3.3.Trust Model: Issuer Onboarding

Instead of permitting every wallet to create certificates, the system limits certificate creation to those addresses pre-approved by a validator list within the smart contract. Following a submission by a prospective institution, current validators engage a voting process via[15] a dedicated Vote () function embedded in the code. Only when each member agrees does integration occur, resulting in inclusion on the official issuer registry. Authority for validation spreads among participants in the group, avoiding reliance on one central authority. Access remains limited strictly to organizations that pass this collective review process[16].

3.4.Core Algorithms

3.4.1. Algorithm 1: Certificate Issuance

- Input: Certificate file c, issuer credentials.
- Step 1: Compute H(c) = SHA-256(c).
- Step 2: Upload c to IPFS and obtain its Content Identifier (CID).
- Step 3: Generate RSA signature S over H(c) using the issuer's private key.
- Step 4: Invoke the smart contract function addCertificate(H(c), CID, S).
- Step 5: Insert H(c) into the Bloom Filter.
- Output: On-chain transaction receipt and hash record.

3.4.2. Algorithm 2: Certificate Verification

- Input: Certificate file c' submitted for verification.
- Step 1: Compute H(c').
- Step 2: Query the Bloom Filter with H(c'). If the filter returns false, the certificate is immediately reported invalid and the process terminates without an on-chain query; if it returns true, proceed to Step 3.
- Step 3: Call the smart contract function verifyCertificate(H(c')).
- Step 4: Compare the on-chain stored hash with H(c') and validate the issuer's RSA signature.
- Output: Valid or Invalid verification result.

3.4.3. Algorithm 3: Issuer Onboarding

- Input: Onboarding request from a new institution.
- Step 1: Existing validators cast votes through the contract's Vote() function.
- Step 2: Unanimous approval is required to proceed.
- Step 3: On approval, the institution is added to the trusted validator list.
- Output: Updated validator consortium.

3.4.4. Algorithm 4: Certificate Correction

- Input: Correction request for a previously issued certificate c.
- Step 1: Confirm the existence of the original certificate via the Bloom Filter and on-chain record.
- Step 2: Mark the original hash as invalid in a revocation list, leaving the original entry intact for audit purposes.
- Step 3: Hash and store the corrected certificate c_new on-chain as a new record.
- Step 4: Insert the new hash into the Bloom Filter.
- Output: Corrected certificate record with a preserved audit trail, without retroactively altering the original blockchain entry.

4. Implementation And Results

Implementation began using Solidity smart contracts on the Ethereum Sepolia testnet. A separate layer

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

handled communication via Node.js with Web3.js support. The interface appeared through React.js, including connection capability for MetaMask. Off-chain files found their location within IPFS storage. Contract logic underwent verification by Mocha combined with Chai assertions. Static checks followed afterward, relying on Slither analysis tools. Testing then progressed across multiple phases - unit examined first, followed by integration, system behavior, and simulated acceptance scenarios. Processes validated included submission of certificates, generation of hashes, recording onto the chain, identification of repeated entries, along with confirmation steps using Bloom Filters prior to execution[17].

4.1.Verification Latency

Before checking the smart contract, using a Bloom Filter cut search times for fake or missing certificates by about 87%. Most bad queries now stop early, avoiding blockchain access altogether. When a certificate is real, confirmation happens on chain, yet full verification stays below two seconds on Sepolia. The process completes quickly even with network checks included.

4.2.Gas Cost

Each smart contract function was adjusted to reduce data stored on chain along with processing needs. A solitary certificate issuance required about 4.57 USD under tested gas conditions, whereas earlier studies noted expenses of 14.32 USD and 8.24 USD for similar setups. Introducing a fresh institution into the system demanded close to 8.97 USD, marginally above certain competing models; this difference arises due to consensus requiring full agreement among participants instead of relying on one central entity, thereby enhancing confidence in authorization despite added expense.As shown in Table 2 Gas Cost Comparison for Core Operations

Table 2 Gas Cost Comparison for Core Operations

Operation	This System	Prior Work
Add certificate	$4.57	$8.24-14.32
Onboard institution	$8.97	-

4.3.Bloom Filter Accuracy

Below 0.5 percent, the observed error level remained when storing up to fifty thousand certificate fingerprints, using five hashing methods chosen through the formula outlined in Section III-B. One megabyte of storage sufficed under these settings. Reliability was maintained throughout testing, avoiding excessive dismissals due to inaccuracies. Performance stayed consistent within acceptable boundaries required for preliminary screening tasks.

4.4.Security Validation

Despite automated checks via Slither, only one minor unused-variable finding emerged - later resolved prior to launch. Zero critical or moderate flaws appeared across audited contract instances. Throughout evaluation phases, certificate hash integrity remained intact without signs of tampering. Every functional scenario executed successfully: submission routines, legitimacy checks, denial of duplicates, Bloom Filter enforcement, and interaction workflows operated as intended. Blockchain-enforced permanence aligned with expected behavior under stress conditions.

4.5.Results

The experimental results demonstrate that the proposed system performs effectively in terms of security, speed, and cost. The Bloom Filter successfully reduces unnecessary blockchain queries by rejecting invalid certificates before on-chain verification. As a result, verification time for forged or non-existent certificates is significantly reduced.
For valid certificates, the complete verification process takes less than two seconds on average. The observed false-positive rate remains below 0.5%, which is consistent with the theoretical Bloom Filter configuration used in the system. Even when a false positive occurs, the certificate undergoes blockchain verification, ensuring that invalid certificates are never accepted as valid. The average cost of issuing a certificate is approximately USD 4.57 because only certificate hashes, digital signatures, and IPFS references are stored on the blockchain. Although institution onboarding incurs a slightly higher cost due to the consensus-based voting process, it provides stronger trust and accountability among participating organizations. Overall, the results indicate that the proposed framework offers a secure,

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

efficient, and cost-effective solution for certificate validation while maintaining data integrity and resistance to tampering.

5. Security Discussion

The proposed system improves certificate security through the combined use of blockchain technology, SHA-256 hashing, RSA digital signatures, and consortium-based issuer validation. Since only authorized institutions can issue certificates, the risk of fraudulent certificate generation is significantly reduced. The onboarding process requires approval from all existing validators before a new institution can join the network. This mechanism helps prevent unauthorized entities from participating in the system and reduces the possibility of Sybil attacks. To preserve privacy, only certificate hashes and related metadata are stored on the blockchain, while the original certificate files remain on IPFS. As a result, sensitive information is not exposed on the public ledger. Any modification to a certificate changes its hash value, making tampering immediately detectable during verification. Furthermore, the immutability of blockchain records ensures that issued certificates cannot be altered or deleted without authorization. Combined with RSA-based signature verification, the system provides strong guarantees of certificate authenticity, integrity, and traceability.

Conclusion

This paper presented a blockchain-based certificate validation framework that combines SHA-256 hashing, RSA digital signatures, IPFS storage, Bloom Filters, and consortium-based issuer validation. The proposed system improves certificate security by preventing unauthorized modifications and enabling reliable verification of academic and professional credentials. By storing only certificate hashes on the blockchain and keeping documents on IPFS, the framework reduces storage requirements and transaction costs while maintaining data integrity and traceability. The Bloom Filter further improves efficiency by filtering invalid certificate requests before blockchain verification. Experimental results on the Ethereum Sepolia testnet showed that the proposed system reduces verification time for invalid certificates by approximately 87%, maintains a false-positive rate below 0.5%, and verifies valid certificates in less than two seconds. These results demonstrate that the framework is secure, efficient, and practical for real-world institutional use. Future work will focus on mobile application support, integration with institutional databases, and the exploration of scalable blockchain networks to further reduce transaction costs and improve system performance.

Acknowledgements

The authors would like to express their sincere gratitude to the Department of Computer Engineering, D. Y. Patil College of Engineering, Akurdi, Pune, for providing the infrastructure, resources, and support required to carry out this work. We would also like to thank our project guides, faculty members, and staff for their valuable guidance and encouragement throughout the development of this project.

References

[1]. R. Pericás-Gornals, M. Mut-Puigserver, and M. M. Payeras-Capella, “Highly private blockchain-based management system for digital COVID-19 certificates,” International Journal of Information Security, vol. 21, no. 5, pp. 1069–1090, Oct. 2022.

[2]. M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, “EduCTX: A blockchain-based higher education credit platform,” IEEE Access, vol. 6, pp. 5112–5127, 2018.

[3]. A. Garba, Z. Chen, Z. Guan, and G. Srivastava, “LightLedger: A novel blockchain-based domain certificate authentication and validation scheme,” IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1698–1710, Apr. 2021.

[4]. U. Rahardja, A. N. Hidayanto, P. O. H. Putra, and M. Hardini, “Immutable ubiquitous digital certificate authentication using blockchain protocol,” Journal of Applied Research and Technology, vol. 19, no. 4, pp. 308–321, Aug. 2021.

[5]. R. F. Ghani, A. A. Salman, A. B. Khudhair, and L. Aljobouri, “Blockchain-based student certificate management and system sharing using Hyperledger Fabric platform,”

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4612 - 4618
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0605

Periodicals of Engineering and Natural Sciences, vol. 10, no. 2, pp. 207–218, Apr. 2022.

[6]. Y. C. Elloh Adja, B. Hammi, A. Serhrouchni, and S. Zeadally, "A blockchain-based certificate revocation management and status verification system," Computers & Security, vol. 104, May 2021, Art. no. 102209.

[7]. S. Mondal, A. Panja, and S. Karforma, "An efficient e-certificate management system in e-learning using blockchain," Scientific Culture, vol. 89, nos. 3–4, pp. 1–5, Apr. 2023.

[8]. U. Rahardja, A. N. Hidayanto, and M. Hardini, "A globalized, immutable digital certificate authentication framework using blockchain," Journal of Applied Research and Technology, vol. 19, pp. 308–321, 2021.

[9]. R. Priyadarshini, R. Pandey, K. C. Ankit, D. Bhandari, B. Khadka, R. K. Barik, and M. J. Saikia, "A faster, integrated, and trusted certificate authentication and issuer validation system based on blockchain," IEEE Access, vol. 13, pp. 27035–27049, 2025.

[10]. S. A. Sultana, C. Rupa, R. P. Malleswari, et al., "IPFS-blockchain smart contracts based framework to reduce certificate frauds," Information, 2023.

[11]. B. M. Nguyen, T. C. Dao, and B. L. Do, "Towards a blockchain-based certificate authentication system," PeerJ Computer Science, 2020.

[12]. A. Alammary, M. Alhazmi, M. Almasri, and S. Gillani, "Blockchain-Based Applications in Education: A Systematic Review," Applied Sciences, vol. 11, no. 3, Art. no. 1184, 2021.

[13]. M. A. Alkhraiji, A. Alhothaily, and M. A. AlZain, "Secure Academic Certificate Verification System Using Blockchain Technology and Smart Contracts," Sensors, vol. 22, no. 18, Art. no. 6921, Sept. 2022.

[14]. S. S. Yadav, A. Kumar, and R. Singh, "Blockchain-Based Framework for Secure Educational Certificate Verification and Management," International Journal of Information Management Data Insights, vol. 3, no. 1, Art. no. 100165, 2023.

[15]. J. Wang, Y. Wu, X. Li, and H. Zhang, "Blockchain-Enabled Digital Certificate Management and Verification Framework for Higher Education Institutions," IEEE Access, vol. 10, pp. 87456–87470, 2022.

[16]. M. S. Hameed, A. Hassan, and A. A. Ahmed, "A Decentralized Blockchain-Based Framework for Educational Certificate Verification," Computers & Security, vol. 118, Art. no. 102728, 2022.

[17]. S. A. Sultana, M. M. Rahman, F. Ahmed, and M. A. Hossain, "Blockchain-Based Secure Academic Certificate Verification System Using Smart Contracts and IPFS," IEEE Access, vol. 12, pp. 45231–45245, 2024.