International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 4, December 2026, pp. 1800~1815
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i4.pp1800-1815

Design framework for higher education certificates using blockchain technology

Salem S. M. Khalifa[1], Ali Mohamed E. Ejmaa[2], Abdulmawla Najih[3], Ismail Aboulqasim Omar Abumilyana[4]
[1]Department of Computer, The High Institute of Science and Technology, Bi'r al Ghanam, Libya
[2]Department of Computer, The Libyan Center for Biotechnology Research, Tripoli, Libya
[3]Department of Computer Engineering, The High Institute of Science and Technology, Garyan, Libya
[4]Department of Computer, The Faculty of Science and Technology, Technical colleges Administration, Ministry of Technical and Vocational Education, Alriyayna, Libya

Article Info

Article history:

Received Oct 5, 2025
Revised May 15, 2026
Accepted Aug 28, 2026

Keywords:

Blockchain technology
Digital certificates
Framework
Sharing process
Verification process

ABSTRACT

Digital academic credentials face persistent threats from credential fraud, while traditional centralized verification systems suffer from single points of failure and administrative delays. Although blockchain technology offers immutability and trust, existing solutions present significant research gaps: (1) direct on-chain storage violates data protection regulations; (2) a lack of decentralized identity governance leaves networks vulnerable to unauthorized; and (3) existing literature lacks formal operational algorithms and empirical benchmarking under scalable workloads. To resolve these gaps, this paper proposes a multi-tiered design framework for higher education certificates anchored in design science research (DSR) methodology. The proposed solution bridges these limitations by implementing a privacy-preserving hybrid architecture that combines off-chain encrypted File System storage with on-chain SHA-256 hash anchoring. To address governance vulnerabilities, W3C Decentralized Identifiers are integrated with a smart-contract Issuer Registry and a global revocation list to enforce strict issuer access control. Furthermore, formal algorithmic protocols are established for system enrollment, issuance, and off-chain encryption. Empirical evaluations conducted using Hyperledger Caliper demonstrate a 100% transaction success rate, a stable throughput of 17.7 TPS, and sub-second verification latencies (10.53 ms), validating the framework's privacy compliance, security, and performance scalability for global academic verification.

*This is an open access article under the **CC BY-SA** license.*

Corresponding Author:

Salem S. M. Khalifa
Department of Computer, The High Institute of Science and Technology
Bi'r al Ghanam, Libya
Email: salemassaid@gmail.com

1. INTRODUCTION

Academic certifications are achievements highly esteemed, as they have a direct impact, positively on their bearers [1]. The traditional methods of verifying and sharing educational certificates have long been plagued by issues of fraud, inefficiency, and lack of transparency [2]. Blockchain technology has emerged as a revolutionary force in the digital world across various industries, including higher education [3]. It is a distributed ledger technology that places a high premium on security, transparency, and immutability of records. It originally developed as the underlying technology for cryptocurrencies like Bitcoin [4]. It operates on a peer-to-peer network, where each participant has a copy of the entire blockchain, ensuring that no single

entity has control over the data. Blockchain technology presents an innovative solution to address the challenges previously mentioned, makes it virtually impossible to tamper with or alter the stored information [5]–[8]. Despite the transition to digital academic credentials, existing verification paradigms remain vulnerable to credential fraud and administrative friction. Traditional centralized systems introduce single points of failure and verification delays, while direct on-chain blockchain deployments suffer from scalability bottlenecks and violate privacy mandates due to public storage of sensitive personal information. Furthermore, existing decentralized solutions often lack formalized issuer identity governance, exposing networks to unverified issuing entities.

To address these gaps, this paper proposes a multi-tiered design framework for higher education certificates using design science research (DSR). The scope of this work focuses on a hybrid architecture combining off-chain interplanetary file system (IPFS) storage, asymmetric encryption, and permissioned on-chain SHA-256 hash anchoring, complemented by W3C decentralized identifiers (DIDs) for identity governance. Specifically, the objectives of this study are to: (1) establish a privacy-preserving credential lifecycle architecture; (2) mitigate unauthorized issuing entities via a smart-contract issuer registry (Reg_I); (3) formalize algorithmic procedures for enrollment, issuance, and encryption; and (4) empirically benchmark system throughput and latency under stress testing.

The main contributions of this research are as follows:

- A multi-tiered design framework (DSR Methodology): Development of a structured, four-layer architecture (Application and user interface, identity & system management, trust engine, and verification) using the DSR methodology specifically tailored for higher education credentials.
- Privacy-preserving hybrid architecture: Integration of off-chain decentralized storage (IPFS) with asymmetric encryption (encrypting certificate artifacts using the student's public key) paired with on-chain cryptographic hash anchoring (SHA-256) to ensure data sovereignty, and scalability.
- Self-sovereign identity and anti-rogue issuer mechanism: Utilization of W3C DIDs for issuers and holders, combined with a smart-contract-backed Issuer Registry (Reg_I) and Global Revocation List to restrict unauthorized issuing entities and handle credential invalidation.
- Formal algorithmic protocols: Definition of step-by-step algorithms, specifically admin/user enrollment (Algorithm 1), certificate issuance (Algorithm 2), and certificate encryption and off-chain storage (Algorithm 3).
- Empirical benchmarking and evaluation: Comprehensive performance evaluation of the prototype using Hyperledger Caliper across *Issue*, *Verify*, and *Mixed* workloads, demonstrating a 100% transaction success rate, a stable throughput of ~17.7 TPS, and verification latencies as low as 10.53 ms. To highlight the main differences and advantages of the proposed scheme, Table 1 provides a detailed comparison between our approach and existing primary credential revocation models used in digital identity and blockchain systems.

Table 1. Comparison of the proposed work with some primary credential revocation models used in digital identity and blockchain systems

Feature / dimension	Centralized / PKI systems	Naive on-chain systems	Standard decentralized models	Our proposed framework
Data Privacy (GDPR/FERPA)	Low (Centralized storage)	Non-compliant (PII on-chain)	Partial (Unencrypted off-chain)	High (Asymmetric Encryption + Off-Chain IPFS)
Identity Governance	Single CA dependency	None / Weak	Basic Wallet Addresses	W3C DIDs + Smart-Contract Registry (Reg_I)
Revocation Capability	Manual / DB Update	Hard to implement	Partial	On-chain Global Revocation List
Single Point of Failure	High Risk	Low Risk	Low Risk	Zero Risk (Decentralized Architecture)
Formalization and Metrics	N/A	Theoretical only	Basic throughput tests	Algorithmic Protocols + Caliper Workload Stress Testing

The blockchain's [9] history has begun since 2008, when Satoshi Nakamoto [4] designed the blockchain as the building block for Bitcoin. Since then, blockchain has become a significant part of providing information security in education, healthcare, financial technology, and even gaming [3], [10].

Blockchain is a term used to refer to a variety of protocols and technologies that make it possible to record information in a way that is difficult to alter, hack, or forge [11]. Blockchain is a form of distributed ledger technology; this means the history of that ledger (the 'truth') and the data controlled thereon is kept across multiple independent entities (blocks), hosted on a common basis by a network of computers (the chain) [3], [11], [12]. Changes to the ledger can be seen by everyone within the network, which is maintained as a whole. These key features make it very suitable for managing the 'lifecycle' of a certificate,

from origination through amendment to verification by a third party, like an employer. Educational systems currently using blockchain.

Several educational institutions and systems around the world have adopted blockchain technology to issue and manage digital credentials. Here are some notable examples,

a) Blockcert: [13] is open-source blockchain project for creating, issuing, viewing, and verifying blockchain-based certificates. Developed by the Massachusetts Institute of Technology aims to provide a secure and tamper-proof way to issue and verify certificates, such as academic credentials, professional certifications, and other verifiable credentials. It enables students and employers to be able to access the Blockcerts platform by the Blockcerts mobile application for verification purposes. The platform uses the bitcoin blockchain as a provider of trust, and hence the data is immutable [14].
b) University of Nicosia: has embraced this technology to reimagine how academic achievements are recorded, shared, and verified. It was one of the first universities to issue blockchain-based certificates. Recording these certificates on the blockchain involves cryptographic signing and hashing, which allows for instant verification of their validity without the need to contact or involve the issuing institution during the verification process [15].
c) OpenCerts:[16] is an open standard for issuing and verifying education credentials through the Ethereum blockchain, primarily developed in Singapore and collaboration among four government technology agencies. It allows educational institutions to issue tamper-proof digital certificates, which can be easily verified by employers and other stakeholders without relying on the institution that issued the certificate instead of submitting the physical copy.
d) TrustED: [17] is a blockchain-based platform that was developed by Kosta Batzavalis in Australia in 2017. It aims to address issues of credential fraud and streamline the verification process for educational qualifications by leveraging the transparency and immutability of blockchain technology. The platform allows for instant verification of credentials, reducing the time and effort required by employers, institutions, and other stakeholders. Students have control over their data and can choose who to share their credentials with, enhancing privacy and security.

2. RELATED WORKS

This section focuses on related research works identified during the detailed literature survey for improving public blockchains for academic certificate certification. Research on the application of blockchain technology in higher education has been gaining significant interest, focusing on areas such as smart contracts, privacy, accreditation processes, and knowledge management. Raimundo and Rosário [18] presented a systematic review of 37 articles on the use of blockchain technology in higher education. It highlights how blockchain is enhancing efficiency, effectiveness, privacy control, and security of data management mechanisms in higher education processes. The potential benefits of blockchain technology in higher education are discussed, along with future research directions. The references of the article cover various applications of blockchain in higher education, such as diploma anti-forgery systems, online quiz schemes, and achievement record systems. Babu *et al*. [19] discussed the limitations of public blockchains for academic certificate certification, such as privacy concerns and scalability issues, and propose a solution using a permissioned blockchain network, specifically Hyperledger Fabric, for the secure issuance and verification of digital academic certificates. This system enhances transparency, security, and reliability by eliminating the need for intermediaries and ensuring tamper-proof records through a structured process involving cryptographic steps. The paper outlines the architecture and processes for managing academic certificates, including issuance and verification phases, and suggests future improvements for better identity management and consensus mechanisms. Al Hemairy *et al*. [20] discussed the application of blockchain technology in education, particularly for the development and verification of digital academic credentials. It highlighted various projects and initiatives, such as Cerberus, Blockcerts, OpenCerts, and EduCTX, which utilize blockchain for secure credential management. The proposed frameworks aim to enhance security and privacy to issue and verify digital academic credentials using the ECDSA algorithm. The research emphasizes the potential of blockchain to improve the integrity and efficiency of academic credentialing processes, supported by various studies and global adoption examples. Ghorab *et al*. [21] introduces PalCert, a blockchain-based system designed to verify and attest academic certificates in Palestine, which aims to improve the current manual processes overseen by the Ministry of Higher Education (MOHE). Solidity was used to write smart contracts, while the proposed framework was evaluated using the Ethereum Ropsten network. The project aims to create an ecosystem based on an open and permissionless blockchain for issuing certificates. The paper suggests future improvements for the proposed approach and be further enhanced to provide more privacy assurances. Fortune *et al*. [22] focus on the problems with academic transcript issuing and authentication in Nigeria by utilizing blockchain's decentralized nature, transparency, and security

features. The research proposed a blockchain-based system that addresses centralized vulnerabilities, inefficiencies in communication, and unverifiability. This system provides an effective, reliable, and scalable method for academic credential authentication. The following Table 2 compares the academic and training platforms described in the sources based on specified criteria.

Table 2. Comparison of the proposed work with some of the most relevant state-of-the-art systems

Platform criteria	EduCTX (Turkanović *et al.*) [23]	Credit verifiable system (Chen *et al.* Conceptual design) [24]	UZHBC (Gresch *et al.* Conceptual design) [25]	IPFS-Blockchain Framework (Sultana *et al.*) [26]	Certificate chain [27]	This work
Verified Documentation	Academic credits (ECTX tokens)	Course credits, homework, and final exams	Higher education diplomas	Academic certificates	Healthcare training certificates (e.g., CITI, HIPAA)	Academic certificates
Verification Method	multisignature addresses and student ID	Comparing document hashes with on-chain records; ECDSA signatures	Automated hash matching via smart contract	Verifying unique content identifiers (CIDs) via Ethereum network	On-chain querying of full PDF files or metadata	Through website
Blockchain	Ark Blockchain Platform	Hyperledger Fabric	Ethereum (Rinkeby testnet)	Ethereum	Ethereum (private network)	Hyperledger Fabric
Consensus Protocol	Delegated Proof-of-Stake (DPoS)	Consortium consensus (no token consumption)	Proof-of-Work (PoW)	Ethereum network (PoW/PoS)	Proof-of-Work (Ethash)	Ethereum network (PoW/PoS)
Decentralized Off-Chain Storage	No (transactions stored on-chain)	No (ledger replicated across peer nodes)	No (hash stored on-chain; original with student)	Yes (uses IPFS for encrypted data)	No (aims to store full certificates on-chain)	Yes (uses IPFS for encrypted data)
IPFS Support	Not mentioned	Not mentioned	Not mentioned	Yes	Not mentioned	IPFS + Layer 2 (Cost-Effective)
Multi-Node Evaluation	Yes (HEIs act as globally distributed nodes)	Yes (evaluated through peer and orderer nodes)	Yes (replicated across a chain of nodes)	Yes (throughput/latency measured across transactions)	Yes (evaluated using 1 and 2 node configurations)	Yes (throughput/latency measured across transactions)
Performance Evaluation	Described as having "fast transactions"	Detailed computation and communication cost analysis	Preliminary evaluation against stakeholder requirements	Detailed throughput and latency analysis via Hyperledger Caliper	Linear scalability and speed testing for PDF uploads	Detailed throughput and latency analysis via Hyperledger Caliper
Open Source	Yes (based on open-source Ark technology)	Yes (Hyperledger Fabric is open source)	Not specified	Not specified	Yes (software is publicly available)	Yes (software is publicly available)

3. METHOD

Designing a blockchain-based framework for higher education certificates involves a systematic approach, encompassing several distinct phases: including the typical framework architecture, the functioning of the detection algorithm, the implementation process, and the methods used to assess the system's performance. The proposed methodology adopts a DSR approach, focusing on the development of a multi-tier blockchain architecture.

3.1. Requirement analysis and design stakeholder identification

3.1.1. Identify the players (actors)

A critical first step is to identify the key participants, or actors, within this blockchain-based system for certificate issuance and management. These actors may originate from within the confines of the system or may exist externally, engaging in various interactions either with the system itself or with each other.

In the context of our blockchain framework for certificates, the identified actors include,

- Assure (Colleges/university) will issue certificates.
- Holder (Students) will receive them.
- Verifier (Employer) will check their validity.

3.1.2. Identify the functional and roles

In this step, the goal is to defining the functional and nonfunctional requirements and operations for the system as the following. Table 3 summarizes the requirements gathered from interviews with Academy of Graduate Studies and Libyan universities which includes the sections: The Student Administration Office, responsible for verification processes (Employer); and the Data Privacy Protection Department, consulted to ensure compliance with legal regulations. For IT-related matters, input was obtained from Informatic of stakeholders, the department that provides IT infrastructure, software, and services for students and staff. Requirements REQ1 to REQ4 pertain to the issuer's conditions-that is, the demands Academy of Graduate Studies and Libyan universities on the system. REQ5 to REQ6 concerns the delivery of certificates in digital format to students. Finally, REQ7 and REQ11 present requirements for companies seeking (Verifier) to verify certificates.

Table 3. Rules defined in access control

	Issuer (Academy of Graduate Studies and Libyan Universities)
REQ1	Authentication and authorization: i. Only authorized university staff (registrar, exam officer) can sign a certificate. ii. Multi-signature or role-based access control (RBAC) on the issuing smart contract. iii. Ability to revoke a certificate (academic misconduct, degree annulment). iv. Maintain a publicly verifiable revocation list or status field on-chain.
REQ2	Certificate creation: i. Define certificate schema (student name, degree, grades, issuance date, unique ID, hash of original document). ii. Generate a cryptographic hash of the certificate metadata. Write certificate hash + metadata URI to blockchain (or sidechain/L2). iii. Write certificate hash + metadata URI to blockchain (or sidechain/L2). iv. Attach university's digital signature (private key).
REQ3	Data privacy & compliance i. Store only minimal data on-chain (hash, status, timestamps); store full certificate off-chain (IPFS, cloud) with encryption. ii. Comply with GDPR/FERPA – allow student data deletion requests where possible.
REQ4	Audit trail Log every issuance, update, and revocation event with timestamp and signer identity.
	Holder (Student)
REQ5	Wallet & key management i. Primary role: Wallet & key management by non-custodial wallet (mobile or web) to store private keys. ii. Ability to Backup & recovery mechanism
REQ6	Certificate reception i. Receive notification when certificate is issued. ii. Verify issuer's signature and certificate integrity on-chain. iii. Request certificate deletion if allowed by law.
	Verifier (Employer)
REQ7	Verification interface i. Enter certificate ID, scan QR code, or receive a shareable link from student. ii. Automatically check Issuer's public key. iii. Automatically check Certificate hash matches on-chain record
REQ8	Batch verification i. Upload multiple certificate IDs or hashes for bulk verification.
REQ9	Tamper detection i. Display clear result: *Valid*, *Revoked*, *Mismatch*, or *Issuer Unknown*. ii. Show original metadata (degree name, graduation date) from off-chain storage.
REQ10	Trust management i. Maintain a trusted issuer registry (whitelist of university DIDs/public keys). ii. Allow custom trust policies.
REQ11	Audit log Log each verification request (timestamp, verifier IP or DID, certificate ID) for employer compliance.

3.1.3. Identify the mathematical variables and system parameters

in order to explain the enrollment and certificate issuing procedures, the following variables and system parameters are defined as shown in Table 4.

3.2. Proposed framework for higher education's certificates

The framework consists of four distinct horizontal layers, each playing a crucial role in the issuance, storage, and verification of academic credentials which are: Application and user interface layer, identity and system management layer, trust engine layer and verification layer as shown in Figure 1, Here is an overview of each layer and the stages within them.

Table 4. Notations and their descriptions

Symbol	Description
U	The universal set of all potential users within the system. This includes individuals who may act as administrators, issuing institutions, certificate holders (students), or verifiers.
A	The subset of U representing authorized system administrators. An administrator a ∈ A possesses elevated privileges for managing user roles and system configurations.
I	The subset of U representing authorized issuing institutions. An institution i ∈ I is authorized to issue academic certificates.
H	The subset of U representing certificate holders. A holder (student) h ∈ H is the recipient of an academic certificate.
V	The subset of U representing verifiers (employers, other educational institutions). A verifier v ∈ V is authorized to verify the authenticity and validity of academic certificates.
DID_u	The DID associated with a specific user u ∈ U. DIDs are globally unique, resolvable, and cryptographically verifiable identifiers that provide self-sovereign control over identity.
DID_i	The DID specifically associated with an issuing institution i ∈ I.
DID_h	The DID specifically associated with a certificate holder h ∈ H.
C	The set of all academic certificates managed by the system.
c:	A specific academic certificate, where c ∈ C.
D_c	The digital data content of certificate c. This typically includes all academic details, personal information of the holder, and other relevant metadata.
h_c	The cryptographic hash of the certificate data D_c.
S_BC	The current state of the blockchain ledger. This ledger stores immutable records of certificate issuance and revocation.
Entry(h_c)	A data structure representing an entry in the blockchain ledger corresponding to a certificate hash h_c. An entry is defined as a tuple (h_c, DID_i, DID_h, T, R, R_reason), where:
T	The timestamp indicating the exact moment the certificate was issued on the blockchain.
R	A boolean flag indicating the revocation status of the certificate (true if revoked, false if not).
R_reason	An optional string providing the reason for revocation, if R is true.
PK_u	The public key associated with a user u. Used for verifying digital signatures.
SK_u	The secret (private) key associated with a user u. Used for creating digital signatures.
Sig (m, SK_u)	A function that generates a digital signature for a message m using the secret key SK_u of user u. This ensures authenticity and non-repudiation.
VerifySig (m, Sig, PK_u)	A function that verifies a digital signature Sig for a message m using the public key PK_u of user u. It returns true if the signature is valid, false otherwise.
Auth_DB	A secure database (which can be centralized or decentralized) used to store user authentication credentials, roles, and other administrative data. This is distinct from the blockchain ledger.
Role(u)	A function that returns the assigned role of a user u within the system ('Admin', 'Issuer', 'Holder', 'Verifier').
Reg_I	A registry (a smart contract or a secure database) that maintains a list of all authorized issuing institutions i ∈ I and their corresponding DID_i.
Reg_A	A registry (a smart contract or a secure database) that maintains a list of all authorized administrators a ∈ A and their corresponding DID_a.

3.2.1. Application and user interface layer

This is the outermost layer, directly interacting with the three primary stakeholders interact: universities (issuers), students (holders), and employers/verifiers. Its primary function is to provide intuitive and accessible interfaces for various credential-related operations via web and mobile applications.

- Credential issuance portal (Web App): This web application is used by universities to create and issue digital certificates. It serves as the primary interface for university administrators to input certificate data (student name, course, grades, degree awarded) in formats JSON or PDF. This portal initiates the process of generating a verifiable credential.
- Digital wallet (Mobile App): Designed for students, this mobile application acts as a secure repository for their verifiable credentials. Students, as holders, receive and store their digital certificates in this wallet. It facilitates interaction with the system and API calls, allowing students to manage and present their credentials as needed.
- Credential verification portal (Web App): This web application is utilized by employers or other verifiers to confirm the authenticity and validity of academic credentials presented by students. Verifiers can submit a credential through this portal to initiate the verification process, receiving a cryptographically valid or invalid result.

3.2.2. Application and user interface layer

This is the outermost layer, directly interacting with the three primary stakeholders interact: universities (issuers), students (holders), and employers/verifiers. Its primary function is to provide intuitive and accessible interfaces for various credential-related operations via web and mobile applications.

- Credential issuance portal (Web App): This web application is used by universities to create and issue digital certificates. It serves as the primary interface for university administrators to input certificate

data (student name, course, grades, degree awarded) in formats JSON or PDF. This portal initiates the process of generating a verifiable credential.

- Digital wallet (Mobile App): Designed for students, this mobile application acts as a secure repository for their verifiable credentials. Students, as holders, receive and store their digital certificates in this wallet. It facilitates interaction with the system and API calls, allowing students to manage and present their credentials as needed.
- Credential verification portal (Web App): This web application is utilized by employers or other verifiers to confirm the authenticity and validity of academic credentials presented by students. Verifiers can submit a credential through this portal to initiate the verification process, receiving a cryptographically valid or invalid result.

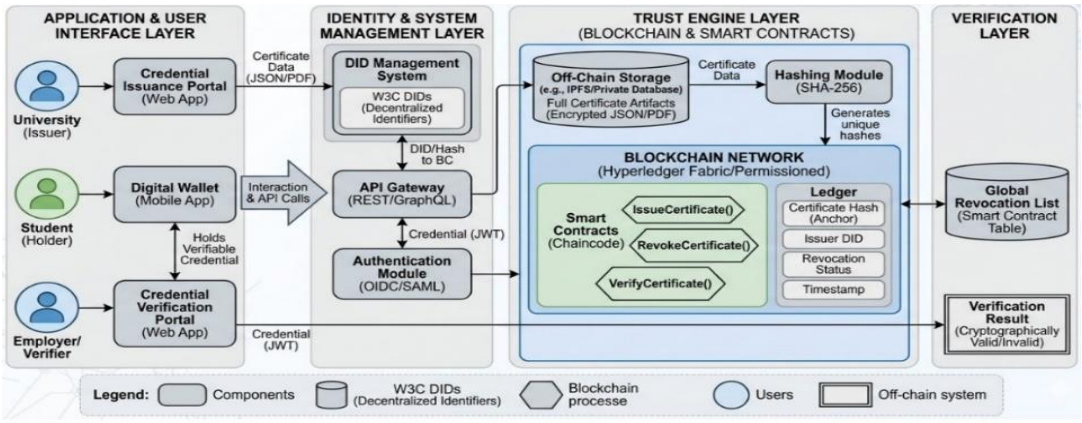


Figure 1. Layered blockchain framework architecture for higher education certificates

3.2.3. Identity and system management layer

This layer acts as an intermediary "Orchestrator" between the user-facing applications and the core blockchain infrastructure. It handles identity management, API interactions, and authentication, ensuring secure and controlled access to the system.

- DID management system: This system is responsible for managing DIDs, which are a new type of identifier that is globally unique, resolvable, and cryptographically verifiable. W3C DIDs are used to represent the identities of issuers (universities) and holders (students) in a decentralized manner. This system facilitates the creation, updating, and resolution of DIDs, often anchoring DID/Hash information to the blockchain (BC) for immutability.
- API gateway (REST/GraphQL): The API Gateway serves as the single-entry point for all interactions and API calls from the Application & User Interface Layer to the underlying services. It supports both REST and GraphQL protocols, enabling flexible and efficient data exchange. Credentials, often in JWT format, are passed through this gateway.
- Authentication module (OIDC/SAML): This module handles user authentication and authorization. It supports standard protocols like OpenID Connect (OIDC) and security assertion markup language (SAML) to securely verify the identity of users (universities, students, verifiers) accessing the system. This ensures that only authorized entities can perform specific actions, such as issuing certificates or initiating verification requests.

3.2.4. Trust engine layer (blockchain and smart contracts)

This is the core of the blockchain system, where the immutability, transparency, and trust mechanisms are enforced. It comprises off-chain storage, a hashing module, and the blockchain network with its smart contracts.

- Off-chain storage: While the blockchain stores immutable proofs, the actual, full certificate artifacts are stored off-chain. This approach is crucial for managing data privacy and handling larger data volumes.
- Hashing module (SHA-256): This module takes the certificate data (from off-chain storage) and generates a unique cryptographic hash using algorithms like SHA-256. This hash acts as a digital fingerprint of the certificate. Any alteration to the certificate data, even a single character, would result in a completely different hash, thus ensuring data integrity and tamper-proofing.

- Blockchain network (Hyperledger Fabric/Permissioned): This is the distributed ledger technology that underpins the entire system. The blockchain network maintains a Ledger that records key information for each certificate, including:
  - Certificate Hash (Anchor): The unique hash of the certificate, serving as an immutable anchor on the blockchain.
  - Issuer DID: The DID of the issuing university.
  - Revocation Status: A flag indicating whether the certificate has been revoked.
  - Timestamp: The time of certificate issuance or last update.
- Smart contracts (Chaincode): These are self-executing contracts with the terms of the agreement directly written into code. In this framework, smart contracts (often referred to as chaincode in Hyperledger Fabric) define the core logic for credential management:
  - IssueCertificate (): A function that records the certificate hash, issuer DID, and other relevant metadata onto the blockchain ledger.
  - RevokeCertificate (): A function that updates the revocation status of a specific certificate on the ledger, effectively invalidating it.
  - VerifyCertificate (): A function that allows querying the blockchain ledger to check the existence and status of a certificate hash.

3.2.5. Verification layer

This final layer is dedicated to the process of verifying the authenticity and validity of a digital certificate, providing a clear outcome to the verifier.

- Global revocation list (Smart contract table): This component, often implemented as a smart contract table on the blockchain, maintains a comprehensive list of all revoked certificates. When a certificate is revoked via the RevokeCertificate () smart contract function, its status is updated here. This list is crucial for ensuring that even if a certificate was once valid, its current status can be accurately checked.
- Verification result (Cryptographically valid/invalid): This is the ultimate output of the verification process. When an employer or verifier submits a credential, the system checks its hash against the blockchain ledger and cross-references it with the Global Revocation List. Based on these checks, a definitive result is provided, indicating whether the credential is cryptographically valid (authentic and not revoked) or invalid (tampered with or revoked). This result provides a high level of trust and assurance in the academic credential.

3.3. Smart contract (Chaincode) development

This section discusses formal mathematical algorithms for two critical processes within a blockchain-based higher education certificate framework: the enrollment of administrators and users, and the subsequent issuance of certificates via the IssueCertificate () function. These algorithms are designed to provide a precise, step-by-step methodology, leveraging cryptographic principles and DIDs to ensure security, authenticity, and integrity.

3.3.1. Algorithm 1 for admin/user enrollment

The first step of this algorithm aims to ensure the allocation of a unique and cryptographically secure identity whether an administrator, an issuing institution, a student (holder), or a verifier. The initial step involves creating a cryptographic key pair: a public key (PK{u_new}) and a corresponding private key (SK {u_new}) by a dedicated key management service.

In the Second step, following key pair generation, a DID_u_new is created for u_new. The CreateDID (PK_u_new) function takes the public key as input and constructs a DID. This DID is a persistent, globally unique identifier that is resolvable to a DID Document, which contains information about how to interact with the DID subject, including their public keys.

In the third step, the user's authentication credentials (Auth_Creds), such as a username and a hashed password, are securely stored in an Auth_DB. This database is distinct from the blockchain and is typically a traditional, secure database managed by the system operator. Auth_Creds offer a more user-friendly authentication mechanism for day-to-day interactions with off-chain services, as well as separation of Auth_DB from the blockchain protects sensitive login information from being exposed on a public ledger.in step four, based on the Role_u_new specified during enrollment, the system assigns the appropriate role to u_new and registers their DID_u_new in the relevant registry. This involves conditional logic:

- If Role_u_new = 'Admin': The DID_u_new is added to Reg_A (Administrator Registry), and the user's role is set to 'Admin' in Auth_DB. This grants u_new administrative privileges within the system.

- If Role_u_new = 'Issuer': The DID_u_new is added to Reg_I (Issuer Registry), and the user's role is set to 'Issuer' in Auth_DB. This authorizes u_new (representing an institution) to issue certificates.
- If Role_u_new = 'Holder' or 'Verifier': The user's role is simply set in Auth_DB. Holders and Verifiers typically do not require registration in a separate DID registry beyond their initial DID creation, as their primary interaction is with their own digital wallet or a verification portal. Role-based access control is critical for maintaining the security and integrity of the system, thus, to prevent unauthorized entities from impersonating legitimate administrators or institutions, the system can cryptographically verify the authority of an entity before allowing them to perform sensitive operations (issuing or revoking certificates). Finally, the algorithm concludes by returning a Success or Failure status, along with the generated DID_u_new, PK_u_new, and SK_u_new. In a real-world implementation, SK_u_new would typically be returned directly to the user's secure client (digital wallet) and not stored by the system itself, where the return of Success indicates that the user has been successfully onboarded and their identity established within the system.

Algorithm 1. For admin/user enrollment

```
Input:
    u_new: The new user to be enrolled.
    Role_u_new: The intended role for u_new ('Admin', 'Issuer', 'Holder').
    Auth_Creds: Authentication credentials for u_new (username, password, or other secure
        authentication tokens).
Output:
    DID_u_new: The Decentralized Identifier generated for u_new.
    PK_u_new, SK_u_new: The public and secret key pair for u_new.
Success or Failure status.
Steps:
1. Generate Key Pair: Generate a new public-secret key pair for u_new:
    (PK_u_new, SK_u_new) ← GenerateKeyPair ()
2. Create DID: Create a new Decentralized Identifier for u_new:
    DID_u_new ← CreateDID (PK_u_new)
    Optionally, anchor DID_u_new or its hash to a public blockchain for global
        resolvability and immutability.
3. Store Authentication Credentials: Securely store Auth_Creds for u_new in Auth_DB.
4. Assign Role and Register DID:
If Role_u_new = 'Admin':
    Add DID_u_new to Reg_A.
    Set Role(u_new) = 'Admin' in Auth_DB.
    Else If Role_u_new = 'Issuer':
    Add DID_u_new to Reg_I.
    Set Role(u_new) = 'Issuer' in Auth_DB.
    Else If Role_u_new = 'Holder' or 'Verifier':
    Set Role(u_new) = Role_u_new in Auth_DB.
    Else: Return Failure (Invalid Role).
5. Return: Return Success along with DID_u_new, PK_u_new, and SK_u_new.
```

3.3.2. Algorithm 2 for issue certificate

This algorithm details the steps involved in issuing a digital academic certificate using the IssueCertificate () smart contract function. It assumes that the issuing institution and the certificate holder h have already been enrolled and possess valid DIDs.The IssueCertificate () function (Algorithm 2) is the core mechanism for creating a verifiable and immutable record of an academic certificate on the blockchain. This algorithm ensures that only authorized institutions can issue certificates and that each certificate is uniquely identified and recorded.

Step 1: Authenticate issuer

Before any certificate can be issued, the system must rigorously authenticate the requesting entity i to confirm it is an authorized issuing institution. This involves two primary checks:

- DID Presence in Reg_I: The system first verifies that the DID_i provided by the requesting institution is present in the Reg_I (Issuer Registry). This confirms that the institution is recognized and authorized to issue credentials within the framework.
- Digital Signature Verification: The institution must digitally sign the request_payload (which contains the details of the certificate to be issued) using its SK_i. The system then uses i's PK_i (obtained from its DID Document) to VerifySig (request_payload, Sig (request_payload, SK_i), PK_i). This cryptographic verification proves that the request genuinely originated from i and has not been tampered with.

Step 2: Generate certificate hash

The digital data content of the certificate, D_c (JSON), is processed by the SHA 256 algorithm, to produce a unique cryptographic hash h_c. The HASH(D_c) function takes the entire digital content and generates a fixed-size string of characters. This hash h_c serves as the immutable digital fingerprint of the certificate. Even a minor change to D_c would result in a completely different h_c. By storing only h_c on the blockchain, the system ensures data integrity and privacy: the sensitive D_c remains off-chain, while its integrity can always be verified against the on-chain hash.

Step 3: Check for existing certificate

Preventing duplicates is essential for maintaining the integrity of the credentialing system and avoiding confusion or potential exploits related to identical certificate hashes, thus, before committing a new certificate record to the blockchain, the system queries the current state of the blockchain ledger (S_BC) to check if an entry with the same h_c already exists.

Step 4: Prepare ledger entry

If the certificate is unique, a new ledger entry (new_Entry) is constructed. This entry is a structured tuple containing h_c, DID_i, DID_h, the current timestamp (T_current), an initial revocationStatus of false, and an empty revocationReason.This step aggregates all the essential, immutable metadata that will be stored on the blockchain. The timestamp provides a verifiable record of when the certificate was issued, and the initial false revocation status is a default that can only be changed by an authorized RevokeCertificate () call.

Step 5: Invoke smart contract

The system then initiates a transaction to call the issueCertificate function of the deployed CertificateManager smart contract on the blockchain network. The CallSmartContract () function transmits h_c, DID_i, DID_h, and T_current as arguments to the smart contract.

Step 6: Update Blockchain State

Upon successful execution of the issueCertificate smart contract function, the blockchain ledger (S_BC) is updated. The new_Entry is added to the ledger, becoming an immutable part of the blockchain's history. This update is propagated across all participating nodes in the network, ensuring consensus and data consistency. Once new_Entry is recorded, it cannot be altered or removed, providing a permanent and verifiable record of the certificate's issuance. This forms the foundation of trust for all subsequent verification processes.

Step 7: Return status

The algorithm concludes by returning a Success status along with the generated h_c. If any step along the way failed (e.g., authentication failure, duplicate certificate hash), a Failure status would be returned.

Algorithm 2. For issue certificate

```
Input:
    i: The issuing institution (i ∈ I).
    h: The certificate holder (h ∈ H).
    D_c: The digital data content of the academic certificate to be issued.
Output:
    •   h_c: The cryptographic hash of the issued certificate.
    •   Success or Failure status.
Steps:
    1   Authenticate Issuer: Verify that is an authorized issuer.
    Check if DID_i is present in Reg_I.
    Authenticate i's request using VerifySig(request_payload, Sig(request_payload, SK_i),
        PK_i).
    If authentication fails, return Failure.
    2   Generate Certificate Hash: Compute the cryptographic hash of the certificate data:
    h_c ← HASH(D_c) (using SHA-256).
    3   Check for Existing Certificate: Query the blockchain ledger S_BC to ensure h_c has
        not been previously issued.
    If Entry(h_c) exists in S_BC, return Failure (Certificate already exists).
    4   Prepare Ledger Entry: Construct the new ledger entry for the certificate:
    T_current ← CurrentTimestamp ()
    new_Entry ← (h_c, DID_i, DID_h, T_current, false, "")
    5   Invoke Smart Contract: Call the issueCertificate function of the CertificateManager
        smart contract on the blockchain network.
    6   Update Blockchain State: Upon successful execution of the smart contract, the
        blockchain ledger S_BC is updated with new_Entry.
    7   Return: Return Success along with h_c.
    CallSmartContract ("CertificateManager", "issueCertificate", h_c, DID_i, DID_h,
        T_current)
```

3.3.3. Algorithm 3 integration and security

A. Certificate encryption and off-chain storage

This stage is critical for balancing data privacy (REQ3) with Immutability. In a blockchain framework, storing large files like PDFs or full JSON objects directly on the ledger is inefficient and expensive. Instead,

our framework uses a Hybrid Approach: the sensitive data is encrypted and stored in a decentralized or secure off-chain environment, while only a unique cryptographic "fingerprint" (the hash) is recorded on the blockchain.

Algorithm 3. Certificate encryption and off-chain storage

```
Input: * D_c: The digital data content of certificate c.
    PK_u: The public key of the certificate holder (student).
    SK_i: The secret (private) key of the issuing institution.
Output: * E\_D_c: The encrypted digital data of the certificate.
    H_c: The cryptographic hash of the certificate data.
    URI: The pointer to the encrypted artifact in off-chain storage.
Steps:
1.  Generate Digital Signature:
    Create a signature of the data to ensure non-repudiation: Sig (D_c, SK_i).
2.  Encrypt Data:
    Encrypt the certificate data using the holder's public key (to ensure only the student
        can decrypt it): E\_D_c \leftarrow Encrypt (D_c + Sig, PK_u).
3.  Generate Immutable Hash (h_c):
    Compute the unique fingerprint: h_c \leftarrow HASH(D_c) using SHA-256.
4.  Off-Chain Storage:
    Upload E\_D_c to Off-Chain Storage.
    Receive the URI representing the storage location.
5.  Anchor to Blockchain:
    Prepare the ledger entry with h_c and URI to be passed to the IssueCertificate ()
        function.
6.  Return:
    Return Success, h_c, and URI.
```

B. Detailed processing steps

i. Data preparation and signing

Before encryption, the Issuing Institution prepares the academic record *D_c*. To ensure non-repudiation (proving who issued the document), the institution signs the data using its private key (*SK_i*).

– Logic: Sig \leftarrow Sig (*D_c, SK_i*).
– Purpose: This signature proves the certificate's origin even after it is encrypted and stored away from the system.

ii. Asymmetric encryption (The privacy layer)

To ensure that only the student (Holder, h) can access the content, the framework uses the student's public key (*PK_u*) for encryption.

– Process: The system encrypts the package (D_c + Sig) using an asymmetric encryption algorithm.
– Result: E_D_c \leftarrow Encrypt (D_c, PK_u). Since it is encrypted with the student's public key, only the student's corresponding private key (SK_h) can unlock it.

iii. Generating the immutable anchor (h_c)

Independent of the encryption, the system runs the original raw data *D_c* through the Hashing Module using the SHA-256 algorithm.

– Property: Even a minor change to the certificate data results in a completely different hash (h_c).
– Role: This hash acts as the "Anchor" on the blockchain ledger (S_{BC}).

iv. Off-chain storage and URI generation

The encrypted file (*E_D_c*) is uploaded to a storage in layer IPFS.

– Output: The storage system returns a uniform resource identifier (URI) or a CID.
– Connection: This URI is the "map" that tells the system where to find the encrypted file when a verifier or student needs it.

v. On-chain anchoring

Finally, the system invokes the IssueCertificate () smart contract. It records:

– The Certificate Hash (*h_c*).
– The Issuer and Holder DIDs (*DID_i, DID_h*).
– The storage URI (to link the on-chain record to the off-chain data).

3.4. Architectural integration and component interaction pipeline

To ensure seamless interoperability, privacy compliance, and performance scalability, the four core components of the proposed architecture operate through a tightly coupled sequential pipeline. W3C DIDs establish cryptographic identity and supply the public key pairs required for payload encryption. The encrypted record is stored off-chain in IPFS, which generates a deterministic CID. This CID, alongside the payload hash, is passed to the smart contract layer (Reg_I), where issuer authority is dynamically validated against the identity registry. Upon successful execution of chaincode logic, Hyperledger Fabric immutably commits the transaction binding {Issuer_DID, Student_DID, IPFS_CID, Hash} to the enterprise

ledger state. This decoupling of identity management (DIDs), off-chain storage (IPFS), business logic (Smart Contracts), and consensus (Hyperledger Fabric) guarantees full GDPR compliance while maintaining sub-second verification latency.

3.5. Prototype implementation

The user interface and issuing dashboard were developed using PHP, HTML5, and JavaScript, enabling issuing institutions to seamlessly register, issue, and manage academic certificates. The core business logic is implemented via smart contracts written in Solidity, leveraging an EVM chaincode execution environment (fabric-chaincode-evm) deployed on the Hyperledger Fabric network. Solidity was chosen for its mature developer tooling and standardized contract structures. During deployment, Solidity source files are compiled using solc into EVM bytecode and ABIs, then installed as chaincode across the Fabric peer nodes. When administrators invoke functions such as IssueCertificate () or RevokeCertificate () via the web interface, transactions are submitted, endorsed according to the channel's endorsement policy, and immutably recorded onto the Fabric ledger under Raft crash-fault-tolerant consensus. The screens to issue and verify are depicted in Figures 2 and 3.

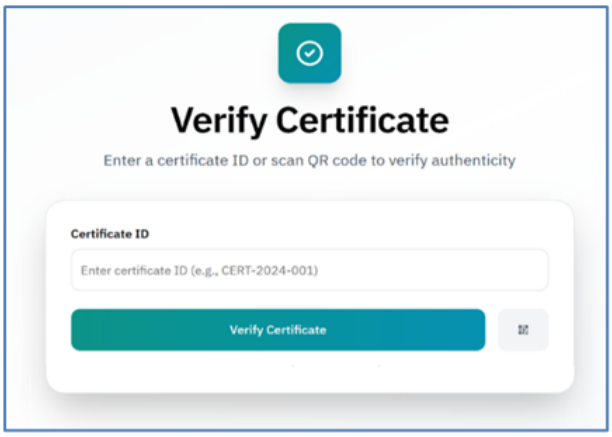


Figure 2. Interface to verify certificate

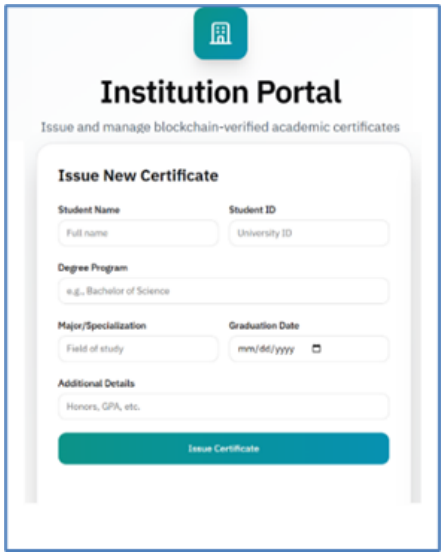


Figure 3. Interface for issue certificate

3.6. Testing and performance evaluation

3.6.1. Platform selection rationale

To select an appropriate underlying ledger for higher education credential verification, public permissionless platforms (such as Ethereum and Solana) and enterprise ledgers (such as Corda and Hyperledger Fabric) were evaluated. Public blockchains were excluded due to volatile gas execution fees, potential GDPR non-compliance regarding publicly visible ledger state, and the absence of native identity governance to prevent unauthorized issuing entities. Hyperledger Fabric was selected due to its enterprise permissioned framework, which integrates membership service providers (MSPs) for strict issuer authentication, eliminates cryptocurrency transaction costs, and supports private data collections (PDCs). Furthermore, its modular Execute-Order-Validate consensus architecture enables sub-second verification latencies and stable throughput required for handling concurrent, multi-institutional credential operations.

3.6.2. Performance evaluation

In this paper, Hyperledger Caliper has been used as a way to evaluate the performance of blockchain platform. From the results shown in the Table 5. The devised experiments were carried out to evaluate the performance of the proposed system in terms of query latency, transaction throughput, resource usage and energy consumption. We observe that the system demonstrates high reliability and stability low and stable latency with full throughput. Where all tests achieved a 100% success rate. The "Verify" workload is the most efficient, with an average latency of only 10.53ms, compared to the "Issue" workload which peaked at 34.77ms. Also, we can note that the system maintained a steady throughput of approximately 17.7 TPS across all test types.

Table 5. Details the performance metrics for three distinct workloads: Issue, verify, and mixed

Round	Workload	Tx Sent	Tx Success	Tx Failed	Success Rate	Send Rate Tps	Throughput Tps	Duration Sec	Latency Avg Ms	Latency Min Ms	Latency Max Ms	Latency P
1	issue	50	50	0	1.0	20	17.76	2.82	34.77	20.0	127.5	31.0
2	verify	50	50	0	1.0	20	17.72	2.82	10.53	4.4	15.3	10.3
3	mixed	50	50	0	1.0	20	17.56	2.85	20.87	2.9	49.9	19.2
4	issue	50	50	0	1.0	20	17.79	2.81	35.41	19.9	130.7	32.0
5	verify	50	50	0	1.0	20	17.66	2.83	10.25	4.5	22.6	10.2
6	mixed	50	50	0	1.0	20	17.36	2.88	21.94	8.5	50.2	20.7

Base on the Figure 4, Colman (X) illustrates the relationship between throughput and latency across different workloads. While throughput remains nearly constant, latency varies significantly, with the Issue workload exhibiting the highest delay due to write overhead. Colman Y presents latency percentiles, where the Issue workload shows high variability (P99 ≫ P50), whereas the verify workload maintains consistent performance across all percentiles. Table 6 presents a comparison for our system with other major blockchain credentialing systems like Blockcerts and OpenCerts.

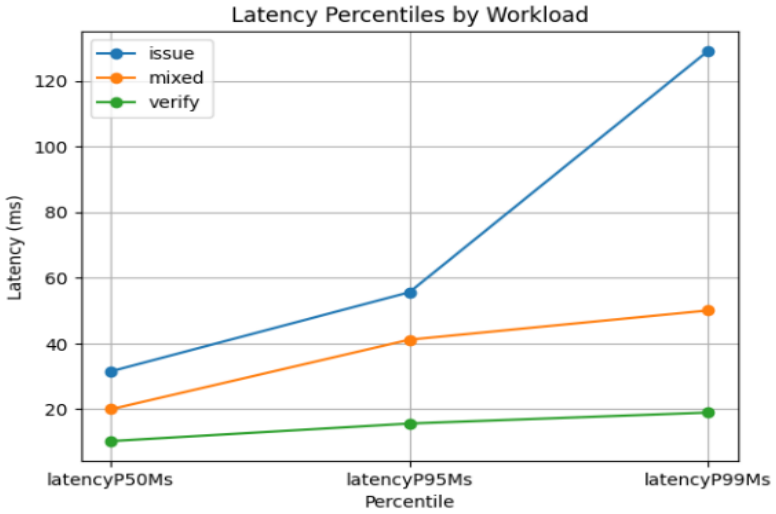


Figure 4. Shows comparing response time distribution across workloads

Table 6. Comparison of blockchain credentialing frameworks

Feature	Our framework	Blockcerts	OpenCerts
Primary Blockchain	Hyperledger Fabric (Permissioned)	Bitcoin	Ethereum
Storage Strategy	Hybrid: Encrypted data on IPFS; Hash on-chain	Typically, local storage or cloud; Hash on-chain	IPFS + Ethereum
Identity Management	W3C DIDs	Public/Private Key pairs	Ethereum addresses / Smart Contracts
Revocation Method	Global Revocation List (Smart Contract Table)	Revocation sub-addresses or Batch Revocation	Smart Contract state updates
Privacy Strategy	Asymmetric Encryption (Holder's Public Key)	Selective disclosure / Hashing	Hashing / Selective disclosure
Verification	Cryptographic hash matching via Web Portal	Blockcerts Mobile App / Web	OpenCerts Web Renderer
Efficiency	High (Low latency ~10ms-35ms)	Low (Limited by Bitcoin block times)	Moderate (Limited by Ethereum gas/congestion)

3.6.3. The advantages of our framework

Based on the Table 5 we can summarized the advantages of our framework

– Data sovereignty and privacy: Unlike some public blockchain implementations that store metadata in the clear, our framework explicitly uses Asymmetric Encryption so that only the student (Holder) can decrypt the certificate content.

- Performance stability: By utilizing Hyperledger Fabric, our framework achieves a steady throughput (approx. 17.7 TPS) and significantly lower latency compared to public chains like Bitcoin, which face scalability and high energy consumption issues.
- Permissioned trust: Our model uses an Issuer Registry (Reg_I) to ensure only authenticated universities can issue credentials, reducing the risk of "rogue" issuers that can exist on completely open public chains.

4. CONCLUSION AND FUTURE WORK

This paper presented a multi-tiered design framework for higher education certificates using DSR methodology to eliminate credential fraud, administrative friction, and single points of failure inherent in traditional credentialing systems. By establishing a four-layered operational architecture spanning application and user interface, identity and system management, trust engine, and verification the proposed solution addresses critical research gaps regarding data privacy regulations, issuer governance, and empirical benchmarking.To satisfy strict privacy mandates , the system implements a hybrid architecture that decouples sensitive credential artifacts from the public ledger, pairing asymmetric off-chain IPFS storage with on-chain SHA-256 cryptographic hash anchoring. Issuer governance and security vulnerabilities are mitigated by integrating W3C DIDs alongside a smart-contract Issuer Registry and an on-chain Global Revocation List to prevent rogue entities from issuing credentials. Furthermore, step-by-step formal algorithmic protocols were established to standardize user enrollment, certificate issuance, and off-chain payload encryption. Empirical performance evaluation conducted via Hyperledger Caliper on a permissioned Hyperledger Fabric network demonstrated 100% transaction success rates, a steady throughput of 17.7 TPS, and sub-second verification latencies as low as 10.53 ms across diverse workloads, validating the framework's stability, low overhead, and scalability. Comparative evaluations against existing platforms confirm that the framework provides superior data sovereignty, zero gas execution costs, and robust decentralized identity controls suitable for higher education institutions. While the prototype successfully validates the framework's theoretical and technical feasibility in controlled environments, future research will focus on expanding interoperability with heterogeneous blockchain networks, conducting large-scale stress testing across higher transaction volumes, integrating the framework with national educational infrastructures (such as Libyan universities and the Academy of Graduate Studies), and developing a production-grade student mobile wallet.

FUNDING INFORMATION

The work was fully self-funded by the authors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Salem S. M. Khalifa	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			✓
Ali Mohamed E. Ejmaa	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Abdulmawla Najih	✓	✓	✓	✓		✓	✓	✓		✓	✓		✓	✓
Ismail Aboulqasim Omar Abumilyana	✓	✓		✓		✓	✓			✓	✓		✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding authors, [Salem S.M Khalifa; Ali Mohamed E. Ejmaa], upon reasonable request.

REFERENCES

[1] T. Healy, S. Côté, J. F. Helliwell, and S. Field, *The well-being of nations: The role of human and social capital*. OECD Publishing, 2001.

[2] F. R. Vidal, F. Gouveia, and C. Soares, "Blockchain application in higher education diploma management and results analysis," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 6, pp. 871–882, 2020, doi: 10.25046/aj0506104.

[3] R. Widayanti, E. P. Harahap, N. Lutfiani, F. P. Oganda, and I. S. P. Manik, "The impact of Blockchain technology in higher education quality improvement," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 2, p. 207, 2021, doi: 10.26555/jiteki.v7i2.20677.

[4] S. Nakamoto, "A peer-to-peer electronic cash system," *Bitcoin*, vol. 4, no. 2, p. 15, 2020, [Online]. Available: https://www.klausnordby.com/bitcoin/Bitcoin_Whitepaper_Document_HD.pdf.

[5] F. Agustin, Q. Aini, A. Khoirunisa, and E. A. Nabila, "Utilization of blockchain technology for management e-certificate open journal system," *Aptisi Transactions on Management (ATM)*, vol. 4, no. 2, pp. 134–139, 2020, doi: 10.33050/atm.v4i2.1293.

[6] Q. Aini, I. Handayani, and F. H. N. Lestari, "Utilization of scientific publication media to improve the quality of scientific work," *Aptisi Transactions on Management (ATM)*, vol. 4, no. 1, pp. 1–12, 2019, doi: 10.33050/atm.v4i1.711.

[7] E. Guustaaf, U. Rahardja, Q. Aini, H. W. Maharani, and N. A. Santoso, "Blockchain-based education project," *Aptisi Transactions on Management (ATM)*, vol. 5, no. 1, pp. 46–61, 2021, doi: 10.33050/atm.v5i1.1433.

[8] K. B. Dubey and M. Goyal, "Smart certificate using blockchain," *Journal of Computer Science*, vol. 18, no. 9, pp. 877–884, 2022, doi: 10.3844/jcssp.2022.877.884.

[9] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business and Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017, doi: 10.1007/s12599-017-0467-3.

[10] S. Chen, X. Liu, J. Yan, G. Hu, and Y. Shi, "Processes, benefits, and challenges for adoption of blockchain technologies in food supply chains: a thematic analysis," *Information Systems and e-Business Management*, vol. 19, no. 3, pp. 909–935, 2021, doi: 10.1007/s10257-020-00467-3.

[11] A. El Nokiti and S. A. M. Yusof, "Exploring the perceptions of applying blockchain technology in the higher education institutes in the UAE," p. 8, 2019, doi: 10.3390/proceedings2019028008.

[12] R. F. Ghani, A. A. Salman, A. B. Khudhair, and L. Aljobouri, "Blockchain-based student certificate management and system sharing using hyperledger fabric platform," *Periodicals of Engineering and Natural Sciences*, vol. 10, no. 2, pp. 207–218, 2022, doi: 10.21533/pen.v10i2.2839.

[13] Philipp Schmidt, "Blockcerts: an open infrastructure for academic credentials on the blockchain," *MIT Media Lab*, 2016, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077545474&partnerID=40&md5=ed1f570f7f21ebb5b1e1a362ffd11334.

[14] Blockcerts, "Recipient experience guide," *Blockcerts.org*, 2024. https://www.blockcerts.org/guide/recipient_experience.html (accessed Aug. 08, 2024).

[15] K. Karasavvas, "Revoking records in an immutable ledger: A platform for issuing and revoking official documents on public blockchains," *Proceedings - 2018 Crypto Valley Conference on Blockchain Technology, CVCBT 2018*, pp. 105–111, 2018, doi: 10.1109/CVCBT.2018.00019.

[16] Ranosys Technologies, "OpenCerts - an easy way to check and verify your certificates," *OpenCerts.io*, 2019. https://www.ranosys.com/blog/insights/opencerts-a-blockchain-based-easy-way-to-check-and-verify-your-educational-certificates-nationwide/ (accessed Aug. 08, 2024).

[17] K. Batzavalis, R. Bala, and A. Norta, "TrustED A Platform for Leveraging Blockchain Technology for the Storage, Issuance and Authentication of Academic Credentials," *TrustED*, 2018. https://ico.trusteducation.io/wp-content/uploads/2018/11/TrustED-Whitepaper.pdf (accessed Dec. 16, 2021).

[18] R. Raimundo and A. Rosário, "Blockchain system in the higher education," *European Journal of Investigation in Health, Psychology and Education*, vol. 11, no. 1, pp. 276–293, 2021, doi: 10.3390/ejihpe11010021.

[19] E. S. Babu, B. K. N. Srinivasarao, I. Kavati, and M. S. Rao, "Verifiable authentication and issuance of academic certificates using permissioned blockchain network," *International Journal of Information Security and Privacy*, vol. 16, no. 1, 2022, doi: 10.4018/ijisp.2022010107.

[20] M. Al Hemairy, M. Abu Talib, A. Khalil, A. Zulfiqar, and T. Mohamed, "Blockchain-based framework and platform for validation, authentication and equivalency of academic certification and institution's accreditation: UAE case study and system performance (2022)," *Education and Information Technologies*, vol. 29, no. 14, pp. 18203–18232, 2024, doi: 10.1007/s10639-024-12493-6.

[21] A. S. Ghorab, R. S. Rasheed, M. S. Salah, and A. A. AbuSamra, "PalCert: A blockchain-based certificate attestation and verification system for HEIs in Palestine," *Technical and Vocational Education and Training*, vol. 39, pp. 123–139, 2024, doi: 10.1007/978-981-99-7798-7_10.

[22] A. C. Fortune, E. J. Garba, U. Mohammed, and A. A. Kadams, "Blockchain-enabled conceptual framework for enhancing academic transcript issuance and authentication in the Nigerian educational system," *International Journal of Development Mathematics (IJDM)*, vol. 1, no. 2, pp. 227–236, 2024, doi: 10.62054/ijdm/0102.19.

[23] 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, doi: 10.1109/ACCESS.2018.2789929.

[24] C. L. Chen, T. Wang, W. J. Tsaur, W. Weng, Y. Y. Deng, and J. Cui, "Based on consortium blockchain to design a credit verifiable cross university course learning system," *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/8241801.

[25] J. Gresch, B. Rodrigues, E. Scheid, S. S. Kanhere, and B. Stiller, "The proposal of a blockchain-based architecture for transparent certificate handling," *Lecture Notes in Business Information Processing*, vol. 339, pp. 185–196, 2019, doi: 10.1007/978-3-030-04849-5_16.

[26] 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 (Switzerland)*, vol. 14, no. 8, 2023, doi: 10.3390/info14080446.
[27] J. Tellew and T. T. Kuo, "CertificateChain: Decentralized healthcare training certificate management system using blockchain and smart contracts," *JAMIA Open*, vol. 5, no. 1, 2022, doi: 10.1093/jamiaopen/ooac019.

BIOGRAPHIES OF AUTHORS



Salem S. M. Khalifa is a lecturer at the Department of Computer Science, The High Institute of Science and Technology, Bi'r al Ghanam. He received the Higher Diploma degree in Programming and Systems Analysis from the Higher Vocational Computer Technologies Institute, Tripoli, Libya in 1995, the B.S. degree in Programming and Systems Analysis / from Ebn. Al. Haytham Higher Technological Education & Scientific Research Center, Tripoli, Libya in 2005, the M.S. degree in Information Technology from University Utara Malaysia in 2010, and the Ph.D. in Science and Technology from University Science Islam, Malaysia in 2017. He can be contacted at email: Salemassaid@gmail.com.



Ali Mohamed E. Ejmaa is a research scholar at the Libyan Center for Biotechnology Research. He received the B.S. degree in computer science from the Faculty of Science, Tripoli University, Tripoli, Libya, in 2003, and the M.S. degree from the Faculty of Computer Science, Universiti Putra Malaysia, Serdang, Malaysia, in 2008. He received his Ph.D. degree from the Faculty of Computer Science, Universiti Putra Malaysia, Serdang, Malaysia, in 2017. His research interests are computer networks and simulation. He can be contacted at email: aliejmaa81@gmail.com.



Abdulmawla Najih was born in 1969 in Gharian, Libya. He received his Bachelor of science degree in computer engineering (hardware) from Sebha University in 1993. He completed his master's and Ph.D degrees in computer engineering at University Putra Malaysia (UPM), Malaysia in 2003 and 2019. His research interests include digital signal processing, biometric recognition, and digital watermarking. He can be contacted at email: nabdulmawla@gmail.com.



Ismail Aboulqasim Omar Abumilyana has been worked as a lecturer at the Department of IT, college of Science and Technology of Alriyayna. He has got the Higher Diploma in Computer of Science from Higher institute and Technology, Benghazi, Libya in 2002, He received M.S degree in information technology project management in 2013 from Teesside University. He interested in desktop application, xml programming, and artificial intelligence. He can be contacted at email: hammamm2012@gmail.com.