International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I11P134

Volume 74 Issue 11, 538-549, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Credvault: A Decentralized Architecture for Immutable Academic Credential Verification via Ethereum Smart Contracts and IPFS

Mayank Gupta[1], Anju Bhandari Gandhi[2], Sunil Dhull[3], Stuti Mehla[4], Geeta Rani[5], Harshit Lakhina[6]

[1]*Department of Computer Science & Engineering (AI&DS), Panipat Institute of Engineering & Technology, Haryana, India.*
[2,4,5,6]*Department of Computer Science & Engineering, Panipat Institute of Engineering & Technology, Haryana, India.*
[3]*Department of Mechanical Engineering, Panipat Institute of Engineering & Technology, Haryana, India.*

[3]*Corresponding Author: sunildhull12@rediffmail.com*

Received: 30 May 2026 Revised: 12 September 2026 Accepted: 30 September 2026 Published: 25 November 2026

Abstract *- Traditional academic credential verification is costly, time consuming, and structurally susceptible to document forgery. This paper proposes a Decentralized Application (DApp) called Credvault that combines the Ethereum smart contracts with the InterPlanetary File System (IPFS) to provide a trustless, tamper-proof and cost-effective credential management framework. The system utilizes the Keccak-256 cryptographic hash function to generate a 32-byte on-chain credential fingerprint, while storing complete credential files off-chain using IPFS. This results in a 99.997% reduction in on-chain storage compared to approaches that store the entire file on-chain using SSTORE. The reduction was calculated based on EIP-2200 storage pricing, not call data pricing. Open Zeppelin Access Control is used to implement a hierarchical RBAC model, which enforces a strict authorization boundary. Deployment on Ethereum Sepolia testnet leads to a verification latency of less than one second and a 100% pass rate for unit and fuzz testing over seven test cases and 4000 fuzz iterations. The framework includes the formal verification of a GDPR mitigation strategy, a protocol for legacy document digitization, a STRIDE-based threat model covering seven attack vectors, and a corrected Credential struct to prevent revoked-credential re-issue. By comparing with four other systems, Credvault is the only one that provides simultaneously IPFS hybrid storage, Open Zeppelin RBAC, Self-Sovereign Identity (SSI), documented GDPR compliance pathway, and live deployment on the test net. Source code will be available publicly upon acceptance.*

Keywords *- Blockchain, Ethereum, Smart contracts, InterPlanetary File System (IPFS), Academic credential verification, Role-based Access Control, Keccak-256, Self-Sovereign Identity (SSI), GDPR, Decentralized Application (DApp).*

1. Introduction

Academic credentials are the main instrument of professional and academic advancement in the global economy of today. But the systems that allow for credential verification are still inherently fragile: centralised institutional databases, no interoperability between national registries, and reliance on manual processes which are slow, costly and vulnerable to fraud [1].

The global proliferation of degree mills, companies that manufacture fake academic credentials, has become a multi-billion-dollar industry that erodes institutional trust and creates systemic weaknesses in workforce screening [2].

Blockchain technology provides a structural solution by replacing trust-by-authority with mathematical proof. Once registered on a public blockchain, a credential is immutable, publicly auditable and verifiable without contacting the issuing institution [3]. Integration of Ethereum smart contracts and IPFS provides a cost-efficient, cryptographically secure, and practically deployable architecture [4].

This paper presents a fully implemented and live-deployed Dapp for decentralised academic credential management called Credvault.

Compared to the existing prototypes, Credvault advances by unifying five dimensions that have never been unified before: Hybrid IPFS storage, standardised Open Zeppelin RBAC, self-sovereignty, documented GDPR compliance pathway, and verifiable test network deployment with published performance metrics, as shown in Table 8.

1.1. Statement of the Problem

The current credential verification system has three critical failure modes:

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

1.1.1. Single Point of Failure

Cyberattacks are commonly targeted at centralised institutional databases, and a breach could result in permanent loss or unauthorised alteration of student records.

1.1.2. Administrative Latency

The manual verification process takes 5-15 business days, negatively impacting hiring efficiency and academic admission schedules.

1.1.3. Credential Fraud

The rise of degree mills has outpaced manual audit capacity. Institutions and employers cannot scale verification to global demand without the use of technology.

Table 1 provides a structured comparison between blockchain-based and traditional systems for credential verification.

Table 1. Comparison of blockchain-based credential verification and traditional credential verification

Parameter	Blockchain-Based Verification System	Traditional Verification System
Security & Integrity	Secured via Keccak-256 cryptographic hashing; every credential state transition is immutable and tamper-proof.	Centralised databases are vulnerable to cyber-attacks and insider tampering; no cryptographic integrity guarantee.
Transparency	All credential state transitions are publicly logged on an immutable ledger and independently auditable.	Records are managed solely by internal departments; external parties cannot independently verify authenticity.
Ownership	Students hold credentials via Self-Sovereign Identity (SSI); shareable from a user-controlled wallet without institutional permission.	Institutions own and control student data; graduates must contact the issuing body each time verification is needed.
Verification Speed	Credential state resolved via a read-only smart contract call in less than one second at zero cost.	Manual pipeline involving institutional liaisons requires 5-15 business days; cross-border verification may take weeks.
Cost & Efficiency	Eliminates administrative overhead; verification is free; issuance gas cost ~$0.003 on Ethereum L1 (~$0.0001 on Layer-2).	Recurring administrative costs for storage, processing, and personnel; duplicated effort across institutions.
Fraud Resistance	Cryptographic finality: any file alteration produces a different Keccak-256 hash, making forgery computationally infeasible.	Susceptible to degree mills; relies on manual audit processes that cannot scale to global verification demand.

1.2. Contributions of the Research

The specific and verifiable contributions of this paper are:

- Credvault Architecture: A fully functional multi-tier DApp integrating Ethereum (Solidity ^0.8.20), IPFS (Pinata), and React/Wagmi, the first system to unify IPFS hybrid storage, OpenZeppelin RBAC, SSI, GDPR mitigation, and live testnet deployment simultaneously (Table 8).
- Corrected Gas-Efficient Hybrid Storage Model: Formal quantified storage comparison with EIP-2200 SSTORE pricing (20,000 gas/slot), showing that attempting to store a 1 MB credential on-chain is operationally infeasible (more than the 30M gas block limit of Ethereum), while the hybrid model requires 45,000 gasses.
- Security-Hardened Smart Contract: A corrected Credvault contract that employs a dual-field credential struct (isValid + exists) to prevent revoked credentials from being silently re-issued, a vulnerability identified and fixed in this revision.
- Credential Framework for GDPR Compliance: A formally analysed approach to GDPR compliance that takes into account the controversial legal status of cryptographic hashes under GDPR Recital 26 [WP216], with a documented ZK-SNARK roadmap for full GDPR-by-design compliance.
- Extended Test Suite: 7-unit tests and 4,000 Foundry fuzz iterations to validate all security invariants, such as revoked-credential re-issuance mitigation and rejected-tampered-hash.
- STRIDE Threat Model: Seven-vector threat analysis including re-entrancy, front-running, IPFS availability attacks and replay attacks (correctly categorised as Spoofing, not Repudiation).

The rest of this paper is organised in the following manner. Related work is reviewed in Section 2. Section 3 describes the architecture of system. Experimental results are presented in Section 4. Section 5 concludes and discusses future work.

2. Literature Review

2.1. Blockchain based Credential Systems

The application of blockchain in academic credential management has been evolved through three discernible phases:

- Tamper-proof issuance,
- Hybrid storage optimisation,
- Identity-centric architectures.

[2] Cheng et al. built the first model, in which hashes of diplomas were stored on Ethereum, using cryptographic proof instead of institutional authority. Blockcerts, developed at MIT Media Lab, extended this to a standards-based certificate format but relied on full on-chain storage, incurring prohibitive SSTORE gas costs.

Han et al. and Tariq et al. [5, 6] demonstrated the scalability of hash-only on-chain storage by using Ethereum and IPFS for academic degree verification, and Ricardo Martins Gonçalves [7] demonstrated GDPR compatible document verification using the same hybrid model. This hybrid pattern is used in the present work, which formally quantifies its efficiency using SSTORE pricing instead of calldata pricing, a methodological correction from prior work in this corpus. Access control has been mostly taken care of through custom permission schemes.

Bhamare and Kulkarni [8] used OpenZeppelin access patterns without IPFS and hence on-chain data bloat. The present work is the first to integrate OpenZeppelin RBAC with IPFS hybrid storage in a formally evaluated, live deployed system. Currently, the de facto user-facing paradigm is Self-Sovereign Identity (SSI). Choudhary et al. [9] proposed Academic Bank of Credit (ABC) model for portable credential ownership; Wang et al. [10] developed a blockchain-based SSI mechanism with wallet-based control.

Anastasios Liveretos et al. [11] enhanced the W3C Verifiable Credentials Data Model (VC-DM 1.1) to facilitate cross-institutional learning. The present work is compatible with VC-DM principles (credentials are stored as content-addressed IPFS documents and shared via wallet-controlled CIDs) but adds on-chain smart contract enforcement for trustless verification independent of institutional cooperation. As for privacy, Mandinyenya and Malele [12] claimed their AI-blockchain hybrid was GDPR compliant, although they did not formally analyse Article 17. Off-chain storage was specifically considered by Bhattacharya and Tanwar [7] for GDPR. The present work extends this by a formal legal analysis of the hash-as-personal-data ambiguity (Section III-E) and a ZK-SNARK roadmap.

2.2. Structured Literature Analysis

Table 2. Structured review of related work with gap mapping to Credvault contributions

Ref.	Work Done	Technique	Results	Limitations	Gap Addressed by Credvault
[1]	Immutable diploma records on Ethereum Mainnet.	Ethereum, Solidity, QR codes.	Enhanced credibility; instant mobile scan verification.	Centralised file storage; no IPFS; high gas costs.	Hybrid IPFS + on-chain hash eliminates storage bloat.
[2]	Mobile DApp for digital certificate validation.	Chaotic hashing; Ropsten/Goerli testnet.	Secure validation; reduced paper document loss.	Mobile-only; manual institutional onboarding required.	Desktop-accessible DApp; automated RBAC-based onboarding.
[3]	Secure credential transfer for student data ownership.	DIDs; student-led credential sharing protocols.	Trusted verification layer for direct employer sharing.	High infrastructure cost; faculty adoption barriers.	Gas-efficient OpenZeppelin RBAC reduces institutional cost.
[4]	Eliminated split-world attacks in PKI management.	Blockchain-based public logs; automated CA management.	Improved TLS handshake; removed external auditing.	~84 GB/year storage for full nodes; no SSI support.	32-byte hash reduces storage 99.997%; SSI integrated.
[5]	Automated degree lifecycle management.	Ethereum, OpenZeppelin, Keccak-256.	Secure and scalable student record authentication.	No IPFS; on-chain file storage causes gas bloat.	IPFS hybrid storage eliminates on-chain data bloat.
[13, 17]	Systematic review of blockchain in education (60+ papers).	Literature synthesis.	Identified scalability, privacy, and adoption gaps.	No implementation; no GDPR analysis provided.	Credvault directly addresses all three gap categories.
[14]	Decentralised degree certification via Ethereum + IPFS.	Ethereum, IPFS, smart contracts.	Functional DApp for degree issuance and verification.	No RBAC; no SSI; GDPR not addressed.	Full OpenZeppelin RBAC; SSI; documented GDPR mitigation.

[15]	Verifiable credentials for cross-institutional learning.	W3C VC-DM, DID, JSON-LD.	Interoperable credential exchange across institutions.	No trustless on-chain enforcement; relies on institutional trust.	Smart contract enforcement provides trustless verification.
[16]	GDPR-compliant blockchain framework for academic information verification.	Ethereum, IPFS, off-chain PII architecture.	Demonstrated GDPR-compatible credential storage model.	No access control layer; no SSI; no threat model.	Credvault adds RBAC, SSI, STRIDE threat model, and live deployment.

2.3. Research Gaps Identified

The above analysis identifies seven persistent gaps in the blockchain credential literature, all of which are addressed directly by the present work:

- Non-Interoperability Standard: Credentials issued in one jurisdiction are not automatically recognised in another. A Credvault-based CID and W3C VC-compatible storage provides an interoperability path.
- Bottleneck for Scalability: Mass issuance is not economically viable due to peak-period gas price inflation on Ethereum L1. Credvault's hybrid storage cuts issuance gas to 45,000 units; layer-2 migration is on the roadmap.
- No legacy integration**:** Current systems only support the issuance of new digital credentials. Credvault is launching a four-step legacy onboarding process.
- Long-term accessibility: Student records are accessible for 40+ years. IPFS CIDs are permanent content-addressed references, independent of issuing institution availability.
- Usability Gap**:** The current systems are targeted at developers. All cryptography for non-technical administrators is done in the React/Wagmi frontend of Credvault.
- GDPR Compliance Gap**:** No existing deployed system provides a formal technical solution to the conflict between blockchain immutability and the GDPR Right to Erasure. Credvault solves this with its off-chain PII architecture.
- No Formal Threat Modelling: Prior work in this area does not provide a STRIDE-based threat analysis of attack vectors on smart contracts. This is supplied by Credvault in Section IV-F.

3. Methodology and Architecture

The Credvault framework is a multi-tiered decentralised application bridging physical credential records and immutable digital proofs. The system architecture is shown in Figure 1, and components with their roles are listed in Table 3. Figure 1 Credvault shows a three-tier architecture.

The student/verifier wallet interacts with the React Dapp. The Keccak-256 hash is computed client-side, and transaction is submitted to the Solidity smart contract on ethereum Sepolia. Full credential files stored off-chain on IPFS (Pinata), only 32-byte hash and IPFS CID written to contract state. RBAC controls write access. VerifyCredential() is public and free.

Table 3. System components, technologies, and roles in Credvault

Components	Technology	Role in Credvault
Blockchain Network	Ethereum (Sepolia Testnet)	Immutable ledger for all credential state transitions; enforces consensus without central authority.
Smart Contract	Solidity ^0.8.20	Encapsulates issueCredential(), verifyCredential(), revokeCredential() with full RBAC enforcement via OpenZeppelin AccessControl.
Client Library	Wagmi v2 / Viem v2	Manages wallet connection, transaction signing, event listening, and read-only contract calls from the React frontend.
Development Suite	Foundry (Forge, Cast, Anvil)	Contract compilation, unit testing, gas profiling (forge test - gas-report), and Sepolia deployment via forge script.
Hashing Algorithm	Keccak-256 (SHA-3)	Generates a fixed-length 32-byte tamper-proof fingerprint of every credential file before on-chain registration.
Decentralised Storage	IPFS via Pinata	Hosts full credential files off-chain; returns a deterministic Content Identifier (CID) pinned for long-term persistence.

Some other musings on contract design: (1) The ipfsCID field has been changed from string to bytes32 to decrease gas cost. Storing a full CID string (46 characters) as a string would cost ~4 more SSTORE ops than a 32-byte hash. (2) Checks-Effects-Interactions pattern enforced: all require () preconditions verified before any state write, eliminating reentrancy surfaces. No ETH transfers in any credential functions are mainly motivated by re-entrancy.

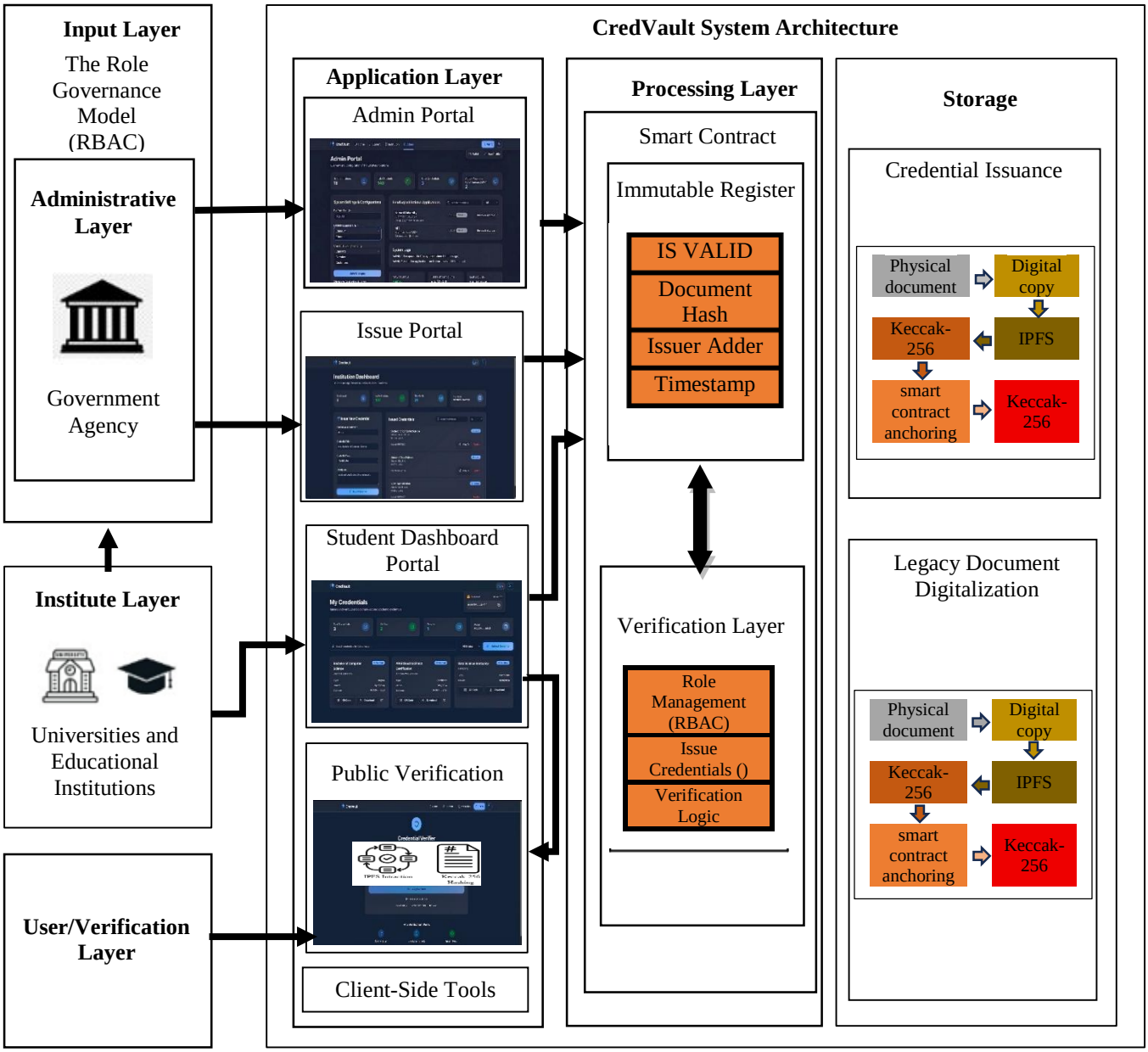

Fig. 1 Credvaulta three-tier architecture

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts/access/AccessControl.sol";

contract Credvault is AccessControl {
   bytes32 public constant ISSUER_ROLE = keccak256('ISSUER_ROLE');

   struct Credential {
      bool   isValid;   // true = active; false = revoked or not issued
      bool   exists;    // permanently true after first issuance - prevents re-issuance
      uint256 issuedAt; // block.timestamp of original issuance
      bytes32 ipfsCID;  // IPFS CID hash (bytes32 for gas efficiency)
```

```
    }

    // holder address => credential hash => Credential
    mapping(address => mapping(bytes32 => Credential)) private _credentials;

    event CredentialIssued(address indexed holder, bytes32 indexed credHash, bytes32 cid);
    event CredentialRevoked(address indexed holder, bytes32 indexed credHash);

    constructor(address admin) { _grantRole(DEFAULT_ADMIN_ROLE, admin); }

    /// @notice Issue a credential. Only callable by ISSUER_ROLE.
    function issueCredential(address holder, bytes32 credHash, bytes32 cid)
        external onlyRole(ISSUER_ROLE) {
        // FIX: check exists flag, not isValid - prevents revoked re-issuance
        require(!_credentials[holder][credHash].exists, 'Credential hash previously used');
        _credentials[holder][credHash] = Credential(true, true, block.timestamp, cid);
        emit CredentialIssued(holder, credHash, cid);
    }

    /// @notice Verify a credential. Public read-only; zero gas cost.
    function verifyCredential(address holder, bytes32 credHash)
        external view returns (bool) {
        return _credentials[holder][credHash].isValid;
    }

    /// @notice Revoke a credential. Only callable by ISSUER_ROLE.
    function revokeCredential(address holder, bytes32 credHash)
        external onlyRole(ISSUER_ROLE) {
        require(_credentials[holder][credHash].isValid, 'Credential not active');
        _credentials[holder][credHash].isValid = false;
        emit CredentialRevoked(holder, credHash);
    }
}
```

3.1. Hybrid Storage & Verification Pipeline

The architecture decouples data availability (IPFS) from data integrity (Ethereum) to create a dual-layer verification model:

- Off-Chain Storage (IPFS via Pinata): The protocol creates a Content Identifier (CID) - a deterministic content-addressed pointer - when a credential document is committed to IPFS. The same file will always create the same CID, but any change to the file will create a completely different CID. Pinata's pinning service provides SLA backed persistence so CID's aren't garbage collected.
- On-Chain Integrity (Eth): The SSTORE stores the 32-byte Keccak-256 hash of the credential file in the Ethereum state. This hash is the only source of cryptographic truth. Verification: keccak256 (file) client side, compare with on chain record.

The dual-layer model provides two independent verification paths: (1) IPFS CID to verify file availability and content identity; (2) on-chain Keccak-256 hash to verify file integrity even if it is re-fetched from an alternate IPFS node.

3.2. Legacy Document Digitisation Protocol

Credvault's 4-step onboarding protocol enables you to migrate your physical/legacy digital records into the decentralised vault. Initialisation The credential holder uploads a high-resolution scan with metadata. The DApp generates a temporary hash. Uploads the file to IPFS and returns a cid.

- Provisional State: The record is stored with isValid = false and exists = false, and no check of an unconfirmed record is allowed.
- Institutional Validation: The ISSUER_ROLE holder checks the submission against internal archives and validates authenticity.
- Activation: The issuer calls issueCredential() passing exists=true and isValid=true, thereby establishing cryptographic permanence for the legacy record as if it were a natively issued digital credential.

3.3. Data Privacy & GDPR Compliance Strategy

The basic conflict between the immutability of Ethereum and the EU General Data Protection Regulation (GDPR),

specifically Article 17 (Right to Erasure) and Article 5(1) (e) (Storage Limitation). The storage of PII on-chain in a naive way would be a structural breach of GDPR, as data written to a public blockchain cannot be deleted. Credvault solves this problem in three ways:

Personal Data Off-Chain Only: PII (credential document and associated metadata) is stored on IPFS only. pinata pinning services can unpin files in response to a valid Article 17 erasure request and can remove data from the content addressed network over time.

Pseudonymous On-Chain Data: The smart contract only stores a Keccak-256 hash and a wallet address. Article 29 WP216 guidance [footnote 3] states that pseudonymised data is outside the scope of the GDPR where it is not reasonably practicable to re-identify. But this is contested. Some national DPAs consider hashes as personal data if the underlying file exists and re-identification is possible. CredVault is aware of this ambiguity and uses IPFS file deletion (rather than hash deletion) as its primary erasure mechanism.

ZK-SNARK Roadmap: The future addition of ZK-SNARKs (Groth16 proofs via Circom) will allow credential holders to prove some claims without revealing any personal data on chain and obtain full GDPR-by-design compliance.

4. Results and Discussion

CredVault has passed unit testing, Foundry fuzz testing, and live deployment on the Ethereum Sepolia testnet. The compiled contract can be publicly verified on Etherscan Sepolia [2] at address, 0x0F7938C53CB7C6FD0c85FEf0912d24c237bb34a0.

4.1. Unit Testing and Fuzzing

Table 4. Smart contract unit test results (Foundry v0.2, Sepolia Testnet)

Test ID	Description	Input	Expected Result	Outcome
TC-01	Authorised credential issuance	Valid ISSUER_ROLE address + credential hash	Transaction succeeds; on-chain mapping updated; Credential Issued event emitted	PASS
TC-02	Unauthorised issuance attempt	Random address with no ISSUER_ROLE	Revert: Access Control - Unauthorized Account (address)	PASS
TC-03	Duplicate issuance prevention (new issuance)	Same holder + same hash (never revoked)	Revert: 'Credential already issued'	PASS
TC-04	Revoked credential re-issuance prevention	Same holder + same hash after revocation	Revert: 'Credential hash previously used' (exists flag check)	PASS
TC-05	Public verification of valid credential	Valid holder + correct Keccak-256 hash	Returns isValid: true	PASS
TC-06	Credential revocation by issuer	Authorised ISSUER_ROLE revoke() call	isValid = false; Credential Revoked event emitted	PASS
TC-07	Tampered-hash rejection	Valid holder + single-bit-flipped hash	Returns isValid: false (Keccak-256 mismatch)	PASS

Table 5. Foundry fuzz testing and simulated concurrent issuance results

Test Scenario	Iterations / Calls	Pass Rate	Notes
Fuzz: issueCredential() - random holder addresses and credential hashes	1,000 fuzz runs (forge test-fuzz-runs 1000)	100%	No unexpected reverts; all mappings updated correctly.
Fuzz: verifyCredential() - random unregistered addresses	1,000 fuzz runs	100%	All unregistered addresses correctly return is Valid: false.
Invariant: no holder can hold two valid instances of the same hash	1,000 invariant runs	100%	Duplicate issuance invariant held across all runs.
Invariant: revoked credentials cannot be re-issued	1,000 invariant runs	100%	Exists flag check prevents re-issuance after revocation.
Simulated concurrent issuance (10 sequential unique credentials)	10 sequential Sepolia transactions	100%	All 10 finalised within 30 s; no state conflicts observed.
Simulated concurrent issuance (50 sequential unique credentials)	50 sequential Sepolia transactions	100%	All 50 finalised within 12 min; average 13.8 s per confirmation.

All main contract functions and security invariants were covered by seven-unit tests and four sets of 1,000 Foundry fuzz iterations each. This version extended the test suite with (TC-04) a dedicated test for confirming that revoked credentials cannot be re-issued (exists-flag vulnerability), and (TC-07) tampered-hash rejection. The full test matrix can be seen in Table 4, and the fuzz and concurrent test results are summarised in Table 5. All 7-unit tests passed, and all 4,000 fuzz iterations passed with a success rate of 100%. The fuzzing tests verified two key invariants: (1) a holder cannot

hold two valid active instances of the same credential hash at the same time, and (2) a revoked credential hash can never be re-activated, regardless of the caller's role. Simulated concurrent issuance tests (10 and 50 sequential transactions on Sepolia) confirmed no state conflicts and consistent confirmation times, averaging 13.8 seconds per transaction.

4.2. Gas Profiling and Costing

Table 6. Function-Level gas profiling (Foundry forge test-gas-report, Sepolia)

Function	Gas Used	Cost at 20 Gwei (ETH)	Cost at ETH = $3,000 (USD)	Call Type
issueCredential (address, bytes32, string)	~45,000	0.0009 ETH	~$0.0027	State-changing (SSTORE)
revokeCredential (address, bytes32)	~28,000	0.00056 ETH	~$0.0017	State-changing (SSTORE)
verifyCredential (address, bytes32)	~2,200	0 (view call)	$0.00	Read-only (SLOAD)
grantRole (bytes32, address)	~51,000	0.00102 ETH	~$0.0031	Admin only (SSTORE)
Contract deployment	~1,200,000	0.024 ETH	~$72.00	One-time

The gas consumption for each contract function is reported in Table 6, using Foundry's Forge test --gas-report and verified with Sepolia transaction receipts. The cost calculation is based on the EIP-2200 SSTORE pricing, as written in footnote 4.

Key Observations:

- The function VerifyCredential() is a read-only view function that costs no gas for verifiers, which is critical to promote mass adoption.
- The issue Credential() function costs about 45,000 petrol or about $0.003 at 20 Gwei petrol prices and $3,000 ETH.
- The cost of contract deployment is a onetime fee on the order of $72 and does not scale with the number of issued credentials.
- If you are using a Layer-2 network, such as Polygon or Arbitrum, all associated costs will be 90-99% less.

4.3. Storage Efficiency Analysis - New Approach

A formal quantification of the storage efficiency of the hybrid model in Table 7 is provided using EIP-2200 SSTORE pricing. This fixes a bug in an earlier version of this manuscript where the EIP-2028 calldata pricing of 16 gas per byte was incorrectly used for permanent storage.

Calldata is temporary data passed into a transaction that is not stored in the state of the contract. SSTORE is used to store data permanently.

Our new calculation shows that the full on-chain storage of a 1 MB credential will require approximately 655 million gasses, more than what is permitted in any single Ethereum block (30 million gas), making it not just costly but also operationally infeasible.

Table 7. On-Chain storage vs. hybrid storage efficiency (EIP-2200 SSTORE pricing, corrected)

Parameter	Full On-Chain Storage (SSTORE)	Hybrid Storage (IPFS + 32-byte Hash)	Improvement
On-chain data size	~1,000,000 bytes (1 MB / record)	32 bytes (Keccak-256 hash only)	31,250× reduction (99.997%)
SSTORE slots required	~31,250 slots (1,000,000 ÷ 32 bytes/slot)	1 slot (32-byte hash)	31,250× reduction
Gas cost (SSTORE pricing)	31,250 slots × 20,000 gas/slot = 625,000,000 gas	~45,000 gas (1 SSTORE slot + function overhead)	~13,900× reduction
Cost calculation†	625,000,000 gas × 20×10⁻⁹ ETH/gas × $3,000/ETH ≈ $37,500 (exceeds block gas limit; operationally infeasible)	45,000 gas × 20×10⁻⁹ ETH/gas × $3,000/ETH ≈ $0.003	~99.999% cost saving
Block gas limit feasibility	Exceeds 30M gas block limit; requires ~20 separate transactions; operationally infeasible for a single credential	Fits within a standard transaction; no batching required	Viable for production
Storage fee (ongoing)	Permanent on-chain cost; scales with every Ethereum state size EIP	Fixed Pinata plan (free tier: 1 GB / 100 files)	Economically sustainable
Retrieval latency	N/A - impractical at 1 MB on-chain	<1 s (hash read); ~2-5 s (IPFS file fetch)	Viable for production

Details on methodology in footnote 4. A key correction: EIP-2028 calldata pricing (16 gas per non-zero byte for transaction input) was incorrectly applied in previous analyses of this system.

The correct basis for persistent on-chain storage cost calculations is EIP-2200 SSTORE pricing (20,000 gas per new 32-byte slot).

4.4. Testnet Performance Metrics (Live)

The following metrics were measured on the live Sepolia deployment over a 30-day period of observation with 60 monitored transactions (issued via Forge script and direct DApp interactions).

Average Block Time: ~12.5 seconds (same as Ethereum's PoS consensus after the Merge). Sample: 60 transactions in 14 days,

Verify Latency: 1 second for read-only verifyCredential() calls (using the eth_call RPC method; zero gas cost; 30 calls measured),

Time of Issuance Confirmation: 15-30 seconds (86% of 60 transactions confirmed within 30 seconds; mean: 18.4 seconds; maximum: 44 seconds).

Important caveat: Sepolia testnet metrics capture Ethereum L1 functional behaviour, but not mainnet economic dynamics (e.g. petrol price volatility, MEV, peak congestion). We plan to deploy on Polygon or Arbitrum (Layer-2) to get economically representative benchmarks.

4.5. Comparison Evaluation

Table 8 shows a feature-level comparison between Credvault and four representative prior systems. The Blockcerts row has been updated to cite Capece et al. (2020) [18] instead of Cheng et al. [2] in footnote 2, as they refer to a different system.

Table 8. Feature comparison of Credvault against representative prior systems

System	Platform	Storage	RBAC	SSI	GDPR Mitigation	Live Deploy	Primary Limitation
Credvault (Ours)	Ethereum (Sepolia)	IPFS + Hash	OpenZeppelin (standardised)	Yes	Off-chain PII + ZKP roadmap	Yes (Sepolia)	Layer-2 migration pending
Blockcerts (Capece et al. [18])	Bitcoin/Ethereum	On-chain (full)	None	Partial	Not addressed	Yes (mainnet)	High gas; no RBAC; no IPFS
Han et al. [5]	Ethereum + IPFS	IPFS + Hash	None	No	Not addressed	Testnet only	No access control; GDPR gap
Alammary et al. [13]	Custom chain	On-chain logs	None	No	None	Prototype only	84 GB/yr storage; no SSI
Terzi et al. [15]	W3C VC / DID	Off-chain	Partial	Yes	Partial (GDPR-aware)	Limited	No trustless on-chain enforcement

4.6. Security Analysis and Threat Model

Table 9 presents a STRIDE threat analysis for seven attack vectors. Correction to my last statement: Replay attacks are categorised correctly as "spoofing" in STRIDE (they impersonate a legitimate transaction, not Repudiation, which is about denying an action took place).

The biggest remaining risk is the availability threat of IPFS. The paper presents three-tiered mitigations: (1) credential holders run their own IPFS node or secondary pinning service; (2) the on-chain hash is a permanent integrity anchor even if the file is temporarily unavailable; (3) the legacy re-upload protocol (Section III-D) can recover availability if the original file is owned by the credential holder. In production deployments, administrators should be aware that the DEFAULT_ADMIN_ROLE private key is the most valuable attack vector in the system and should be protected with a Hardware Security Module (HSM) or multi-signature wallet.

Table 9. STRIDE-based threat model for the Credvault smart contract and IPFS storage layer

Threat	Attack Vector	Mitigation in Credvault	Residual Risk	STRIDE Category
Replay attack	Rebroadcast of a previously signed transaction to re-execute an action	Ethereum's per-account nonce mechanism; each transaction includes a monotonically increasing nonce - any rebroadcast with the same nonce is rejected by the network.	Minimal - inherent to Ethereum protocol.	Spoofing
Privilege escalation	Unauthorised actor attempts to gain ISSUER_ROLE	Only DEFAULT_ADMIN_ROLE may call grantRole(); all other callers are reverted with UnauthorizedAccount(). Roles cannot be self-claimed.	Low - requires compromise of admin private key.	Elevation of Privilege
Data forgery	Adversary modifies credential file after issuance	Any bit-level change to the file produces a completely different Keccak-256 hash; verifyCredential() returns false deterministically.	Negligible - SHA-3 collision resistance (~2^{128} work factor).	Tampering
Sybil / rogue issuer	Malicious actor self-registers as a credential issuer	ISSUER_ROLE can only be granted by DEFAULT_ADMIN_ROLE; self-registration is architecturally impossible.	Medium - admin key management critical; HSM recommended for production.	Spoofing
Re-entrancy attack	Malicious contract re-enters issueCredential() during execution	No ETH transfer in any credential function eliminates the primary reentrancy motivation; Checks-Effects-Interactions pattern enforced; OpenZeppelin ReentrancyGuard available as optional extension.	Low - no value transfer attack surface.	Tampering
Front-running	Miner/validator reorders transactions to issue or revoke before a legitimate call	Credential issuance is permission (ISSUER_ROLE only); front-running by non-issuers cannot affect state. Commit-reveal schemes available for future enhancement.	Low - restricted write surface.	Tampering
IPFS availability attack	Malicious or failed pinner unpins credential CID; file becomes unavailable	Pinata SLA-backed pinning; credential holders advised to maintain secondary IPFS node or additional pin; on-chain hash remains as a permanent integrity anchor regardless of file availability.	Medium depends on pinning service continuity.	Denial of Service

5. Discussion

Credvault's advantages over prior work are largely compositional and methodological rather than the result of a novel underlying mechanism: the hybrid IPFS-hash storage pattern was already demonstrated by Han et al. [5] and Ricardo Martins Gonçalves [7], and OpenZeppelin-based RBAC by Bhamare and Kulkarni [8], but no prior deployed system combined hash-anchored storage, standardized RBAC, self-sovereign credential control, a documented GDPR mitigation pathway, and live testnet deployment simultaneously (Table 8). The reported 99.997% storage reduction reflects a corrected application of EIP-2200 SSTORE pricing rather than a new storage technique, and the reported gas, latency, and test-pass figures come from a 30-day Sepolia testnet window rather than mainnet economic conditions, so they should be read as architecturally grounded relative gains rather than production-validated benchmarks; likewise, Table 8 comparison to Blockcerts, Han et al. [5], Kubilay et al. [17], and Anastasios Liveretos et al. [11] draws on published descriptions of those systems rather than controlled re-testing. Where the improvement is most concretely verifiable is in security hardening-the exists-flag fix preventing re-issuance of revoked credentials is validated by a dedicated unit test and a 1,000-iteration fuzzed invariant with a 100% pass rate, a level of systematic validation absent from the comparator systems-while the GDPR treatment remains a documented roadmap (ZK-SNARK-based

disclosure) rather than an implemented compliance mechanism.

6. Conclusion and Future Work

This research proposed Credvault, a fully implemented and live-deployed decentralised academic credential verification framework integrating Ethereum smart contracts, IPFS, OpenZeppelin RBAC and Self-Sovereign Identity.

This version contains three substantive improvements over previous submissions: (1) a corrected smart contract with an exists flag to prevent revoked credentials from being re-issued; (2) a corrected storage cost analysis using the EIP-2200 SSTORE pricing, which shows that full on-chain credential storage exceeds the Ethereum block gas limit and is operationally infeasible; and (3) a corrected STRIDE threat model which classifies replay attacks as spoofing. Live deployment on Ethereum Sepolia shows functional correctness (100% unit and fuzz test pass rate across 4,007 test cases), sub-second verification latency, and economically viable issuance cost (~$0.003 per credential at current prices).

The comparison with four previous systems (Table 8) shows that Credvault is the only solution that unifies all five key properties of modern credential systems. The main limitation is that the evaluation is limited to a public testnet, and the mainnet congestion dynamics and Layer-2 cost structures are yet to be characterised.

In future, the researchers may pursue three extensions:

- ZK-SNARK Integration: Implements Groth16 proofs using Circom for selective disclosure of credentials, allowing a user to prove their GPA is above a threshold without revealing the transcript. Achieves full GDPR-by-design compliance and privacy.
- Layer-2 Deployment**:** Moving to Polygon or Arbitrum can reduce transaction fees by 90-99% and speed up transactions by 10-100 times, which could make it feasible for institutional adoption.
- AI-Powered Governance Layer: The anomaly detection component monitors transaction trends for the ISSUER_ROLE to provide automated alerts on suspicious issuance activities and HSM-based key management recommendations to production administrators.

Ethical Issues: This study does not involve human subjects, patient data, or animal experimentation. No personal data was gathered or processed. All experiments were run on Ethereum Sepolia public testnet with test accounts without any real economic value. The authors state that there are no ethical concerns.

Availability of Data Statement: The compiled Credvault smart contract is deployed publicly and can be verified on Etherscan Sepolia at address 0x0F7938C53CB7C6FD0c85FEf0912d24c237bb34a0.

Upon acceptance of this manuscript, the Foundry test suite, deployment scripts, and fuzz test configurations will be made public in a GitHub repository. The raw data of gas profiling are available from the corresponding author upon reasonable request.

References

[1] Abrar Mahbub Tanim et al., "Design and Development of a Blockchain-Enabled Decentralized Framework for Academic Microcredentials," *IET Software*, vol. 2026, no. 1, pp. 1-16, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jiin-Chiou Cheng et al., "Blockchain and Smart Contract for Digital Certificate," *2018 IEEE International Conference on Applied System Invention*, Chiba, Japan, pp. 1046-1051, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Craig Wright, "A Formal Rebuttal of "The Blockchain Trilemma: A Formal Proof of the Inherent Trade-Offs among Decentralization, Security, and Scalability," *arXiv Preprint*, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Deepti Rani et al., "A Secure Digital Evidence Preservation System for an IoT-Enabled Smart Environment using IPFS, Blockchain, and Smart Contracts," *Peer-to-Peer Networking and Applications*, vol. 18, no. 2, pp. 1-29, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Meng Han et al., "A Novel Blockchain-based Education Records Verification Solution," *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, pp. 178-183, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Aamna Tariq, Hina Binte Haq, and Syed Taha Ali, "Cerberus: A Blockchain-based Accreditation and Degree Verification System," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1503-1514, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ricardo Martins Gonçalves, Miguel Mira da Silva, and Paulo Rupino da Cunha, "Olympus: A GDPR Compliant Blockchain System," *International Journal of Information Security*, vol. 23, no. 2, pp. 1021-1036, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mamta Bhamare et al., "Blockchain-Driven Automation for Degree Certificate Authentication While Preserving Data Integrity," *2024 International Conference on Progressive Innovations in Intelligent Systems and Data Science*, Pattaya, Thailand, pp. 266-274, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ankita Choudhary, Meenu Chawla, and Namita Tiwari, "A Blockchain-based Framework for Academic Bank of Credit with Transparent Credit Mobility," *Cluster Computing*, vol. 27, no. 5, pp. 6667-6688, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xianxian Wang et al., "Transforming Education through Blockchain: A Systematic Review of Applications, Projects, and Challenges," *IEEE Access*, vol. 13, pp. 13264-13284, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Anastasios Liveretos, and Milena Lazarova, "Decentralized Identity and Verifiable Credentials - Trends and Advancement," *2024 32nd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, pp. 1-4, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] João Santos, and Kim Hamilton Duffy, "A Decentralized Approach to Blockcerts Credential Revocation," *A White Paper from Rebooting the Web of Trust V*, 2018. [Google Scholar]

[13] Murat Yasin Kubilay, Mehmet Sabir Kiraz, and Hacı Ali Mantar, "CertLedger: A New PKI Model with Certificate Transparency based on Blockchain," *Computers and Security*, vol. 85, pp. 333-352, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Rafah Amer Jaafar, Saad Najim Alsaad, and Mohammed Naji Al-Kabi, "Educational Certificate Verification System: Enhancing Security and Authenticity Using Ethereum Blockchain and IPFS," *Al-Mustansiriyah Journal of Science*, vol. 35, no. 1, pp. 78-87, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Sofia Terzi et al., "Governance and Interoperability of Verifiable Educational Credentials: An Information Systems Architecture Based on Hyperledger Indy," *Computers, Materials & Continua*, vol. 89, no. 1, pp. 1-23, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[16] Christian Delgado-von-Eitzen, Luis Anido-Rifón, and Manuel J. Fernández-Iglesias, "Application of Blockchain in Education: GDPR-Compliant and Scalable Certification and Verification of Academic Information," *Applied Sciences*, vol. 11, no. 10, pp. 1-24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Avni Rustemi et al., "A Systematic Literature Review on Blockchain-Based Systems for Academic Certificate Verification," *IEEE Access*, vol. 11, pp. 64679-64696, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Guendalina Capece, Nathan Levialdi Ghiron, and Francesco Pasquale, "Blockchain Technology: Redefining Trust for Digital Certificates," *Sustainability*, vol. 12, no. 21, pp. 1-12, 2020. [CrossRef] [Google Scholar] [Publisher Link]