Attestify

A Decentralized Academic Credential Platform

Project Proposal

Supervisor

Mr. Faizan Saleem

Submitted by

Zain Ul Abidin (UOC-CS-F2022-036)

M. Saad Feroz (UOC-CS-F2022-034)

1. Introduction

The increasing rate of academic certificate forgery and fraudulent credential claims necessitates a secure, verifiable, and tamper-proof solution. Traditional certificate verification relies on manual checks or siloed centralized databases, which are often slow, insecure, and susceptible to data breaches or manipulation.

This project proposes a hybrid blockchain-based system designed to ensure the authenticity, integrity, and transparency of academic credentials. It combines the immutable and decentralized trust of a public blockchain with a secure, user-friendly management portal.

By recording cryptographic "fingerprints" of certificates on a decentralized ledger, institutions, employers, and students can instantly and confidently verify legitimacy without the delays and risks of traditional methods.

2. Problem Statement and Objectives

Educational institutions issue thousands of certificates annually, but traditional paper and digital-only verification methods are inefficient and vulnerable to sophisticated forgery. There is a critical need for a system that is tamper-proof, transparent, and automated.

The core objectives of this project are to:

1. **Design and develop a hybrid platform** that separates the user management layer from the decentralized verification layer.
2. **Ensure credential immutability** by storing cryptographic hashes (SHA-256) of all certificates and transcripts on the Ethereum Sepolia testnet.
3. **Provide decentralized and resilient file storage** by storing the actual (and optionally encrypted) certificate files on IPFS (InterPlanetary File System).
4. **Deliver a user-friendly portal** (via an Express.js backend) for institutions, students, and employers to securely issue, manage, and verify credentials.
5. **Drastically reduce fraud** by providing a single, publicly verifiable source of truth for certificate authenticity.

3. System Architecture

Attestify is designed as a hybrid system to leverage the best of both centralized and decentralized technologies, prioritizing both security and user experience.

3.1. The Hybrid Model

The system is composed of two primary layers:

1. **The Decentralized Trust Layer (Blockchain + IPFS):** This layer is the "source of truth" and is fully decentralized.

- **Ethereum Sepolia Testnet:** A public Ethereum test network that stores the immutable *proof*. The smart contract on Sepolia stores a mapping of each student's ID to the cryptographic hash (SHA-256) of their certificate and the IPFS address (CID) where the file is located.

- **IPFS (Inter-Planetary File System):** A decentralized storage network that stores the actual certificate/transcript file (e.g., a PDF). This ensures the file itself is not held on a single server, making it resilient to censorship and data loss.

2. **The Centralized Management Layer (Web App + Backend):** This layer provides the necessary user interface and business logic for usability.

- **React.js:** A user-friendly interface for all roles.

- **Node.js/Express.js:** This server acts as the management hub. It handles user authentication, profile management, file encryption (before uploading to IPFS), audit logging, and email notifications. It facilitates interaction with the blockchain but does not store the official certificates or verification logic.

3.2. Core Workflows

Issuance Workflow:

1. An institution administrator logs into the portal.
2. They upload the student's certificate (e.g., PDF).
3. The backend server generates a SHA-256 hash (a unique fingerprint) of the file.
4. The server (optionally) encrypts the file to protect student privacy.
5. The server uploads the file to IPFS, receiving a unique Content ID (CID).
6. The backend triggers the issueCertificate() function on the smart contract, storing the SHA-256 hash and the IPFS CID on the Ethereum Sepolia blockchain, linked to the student's ID.

Verification Workflow:

1. A third-party verifier visits the public portal (no login required).
2. They either scan a QR code or drag-and-drop a certificate file.
3. The system calculates the SHA-256 hash of the uploaded file *in the browser*.
4. It queries the smart contract (using the student ID from the QR code or file metadata) to retrieve the *official* hash stored on the blockchain.
5. The system compares the two hashes.

- **Match:** The certificate is authentic.
- **No Match:** The certificate is fraudulent or has been tampered with.

4. Technology Stack

Component	Technology	Role
Frontend	React.js / Next.js	Provides the user interface for all roles.
Backend	Node.js / Express.js	Manages user authentication, APIs, and business logic.
Database	MongoDB / PostgreSQL	Stores user information, profiles, and non-critical audit logs. Does not store official certificates.
Blockchain	Solidity (Smart Contract)	Deployed on the Ethereum Sepolia Testnet to manage credential records.
Decentralized Storage	IPFS (Inter-Planetary File System)	Stores the actual certificate and transcript files in a decentralized, resilient manner.
Blockchain Client	Ethers.js	Facilitates communication between the backend and the Ethereum smart contract.
Wallet	MetaMask	Used by the institution's admin account to pay gas fees for issuing/revoking certificates.
Hashing	SHA-256 / Keccak256	Generates unique, tamper-proof digital fingerprints for files and on-chain data.

5. Core Features

5.1. For Institutions (Admin Role)

- **Secure Credential Issuance:** Issue single or batch certificates and transcripts, which are automatically hashed, stored on IPFS, and recorded on the blockchain.

- **Credential Revocation:** An admin-only revokeCertificate() function in the smart contract to mark credentials as invalid (e.g., if issued in error).
- **Administrative Dashboard:** A comprehensive dashboard to view audit trails of all issued credentials, track gas/cost metrics, and view system statistics.

5.2. For Students

- **Personal Credential Profile:** A secure, dedicated profile where students can view all their issued certificates and transcripts (fetched directly from IPFS).
- **QR Code & Sharing:** Generate a unique QR code or a shareable link for any certificate, enabling one-click verification by employers.
- **Email Notifications:** Automatic email notifications upon successful certificate issuance.

5.3. For Verifiers (Employers / Public)

- **Instant Verification Portal:** A simple, public, drag-and-drop interface to verify a certificate's authenticity.
- **QR Code Scanning:** Instantly verify a credential using any smartphone camera.
- **Transparent Proof:** The verification result provides a direct link to the transaction on a public block explorer (like Sepolia Etherscan) for ultimate transparency.

5.4. Security & Data Integrity Features

- **User Authentication & Authorization:** Secure, role-based access control (Admin, Student, Verifier) managed by the backend using JWT (JSON Web Tokens) to protect sensitive actions like issuance and revocation.
- **Secure Hashing (SHA-256):** Guarantees data integrity by creating a unique, tamper-proof digital fingerprint of all certificate files. This hash is stored on-chain to act as the single source of truth.
- **Transcript Integration:** Allows for the separate issuance and verification of academic transcripts alongside final certificates, with each having its own hash and IPFS record for comprehensive academic validation.

6. Implementation Steps

1. **Environment Setup:** Configure development environment with Node.js, React, Hardhat (for smart contract development), and an IPFS client (or a pinning service like Pinata).
2. **Smart Contract Development:** Develop and test the Solidity smart contract (functions: issueCertificate, revokeCertificate, getCredentialHashes).
3. **Deployment:** Deploy the smart contract to the Ethereum Sepolia Testnet.
4. **Backend Development:** Build the Express.js API for user authentication (JWT), profile management, file encryption, and IPFS uploading.
5. **Blockchain Integration:** Integrate the backend with the smart contract using Ethers.js to call its functions.
6. **Frontend Development:** Build the React.js frontend with separate portals for Admin, Student, and the public Verifier.
7. **Feature Implementation:** Implement core features: file upload, hashing, IPFS integration, QR code generation, and the verification logic.

8. **Testing:** Conduct thorough end-to-end testing, including security audits of the smart contract and backend.

7. Expected Outcomes

- A system where issued certificates are fully tamper-proof.
- Instant, free, and reliable verification for employers, removing administrative bottlenecks.
- Full data ownership and control for students over their own credentials.
- Enhanced transparency and trust between institutions and the public.
- A significant reduction in administrative overhead and the risk of certificate fraud.

8. Conclusion

This system provides a secure, transparent, and highly efficient solution to the pervasive problem of certificate forgery. By leveraging the immutability of the Ethereum Sepolia testnet for proof and the resilience of IPFS for storage, it creates a fully decentralized and verifiable record of academic achievement.

This project's pragmatic hybrid architecture ensures that the system is not only powerful and secure but also accessible and user-friendly, delivering a practical, real-world application of blockchain technology that is scalable and ready for institutional implementation.

References

1. Verifi-Chain: A Credentials Verifier using Blockchain and IPFS

   https://arxiv.org/pdf/2307.05797

2. Hybrid Blockchain-based Academic Credential Verification System (B-ACVS)

   https://www.researchgate.net/publication/370292475_Hybrid_Blockchain-based_Academic_Credential_Verification_System_B-ACVS

3. Educational Certificate Verification System: Enhancing Security and Authenticity using Ethereum Blockchain and IPFS

   https://iasj.rdd.edu.iq/journals/index.php/IASJ/article/download/7498/4383

4. Decentralized Document Version Control using Ethereum Blockchain and IPFS

   https://repository.uwl.ac.uk/id/eprint/5926/1/Arshad_etal_CEE_2019_Decentralized_document_version_control_using_ethereum_blockchain_and_IPFS.pdf

5. IPFS-Blockchain Smart Contracts Based Conceptual Framework to Reduce Certificate Frauds in the Academic Field

   https://www.mdpi.com/2078-2489/14/8/446/pdf