Immortal Certificates on Ethereum Blockchain with Off-Chain IPFS storage

Aisha Lalli
New York University
al8211@nyu.edu

Leandro R. Maciel
New York University
lrm371@nyu.edu

Aspen Olmsted
Wentworth Institute of Technology
olmsteda@wit.edu

Abstract

The use of digital certificates is crucial for verifying the authenticity of various credentials in our digital age. However, traditional digital certificate systems suffer from centralization, vulnerability to tampering, and the risk of loss. This paper presents an approach to issuing and managing certificates as Non-Fungible Tokens (NFTs) on the Ethereum blockchain, ensuring their immutability and perpetual existence. The proposed system aims to overcome the limitations of traditional methods by utilizing decentralized storage through the InterPlanetary File System (IPFS) and implementing a robust incentive mechanism for voting and Proof of Stake; the goal is to achieve true permanence and enhanced security for digital certificates. A first phase of a smart contract and front-end interface was achieved, to preserve diplomas and provide a reliable method for verifying their authenticity for employers and other stakeholders. This paper is part of an ongoing effort to develop a robust system, create a prototype to validate the perpetuity assumption, and propose improvements for a global, enduring repository of certificates.

1. Introduction

Digital certificates play a critical role in verifying the authenticity of credentials such as academic degrees, professional certifications, and various licenses [1]. However, traditional digital certificate systems are centralized, making them vulnerable to tampering, loss, and unauthorized access. Additionally, these systems are susceptible to forgery, as external actors can create fake certificates that appear legitimate. Blockchain technology offers a promising solution with its decentralized and immutable ledger, which can be leveraged to issue certificates as Non-Fungible Tokens (NFTs) on the Ethereum blockchain.

This paper introduces “Immortal Certificates,” a system designed to ensure the perpetual existence and invulnerability of digital certificates, addressing the shortcomings of current methods. The goal is to achieve immutability, permanence and enhanced security for digital certificates by utilizing Ethereum blockchain platform, decentralized storage through the InterPlanetary File System (IPFS) and implementing a robust incentive mechanism, for voting on the ingress of new issuers and sustainability via Proof of Stake. Given the fast-paced evolution seen in cybersecurity, new systems should always be designed being acutely aware of the potential vulnerabilities that can arise. Thus, to build a system with a secure-by-design approach is the aim, considering possible points of failure and proactively addressing them. The current work is part of an ongoing effort to refine and enhance the reliability and security of digital certificates in a digital age.

2. Background

Blockchain is a revolutionary technology that has exploded in popularity over the years due to its ability to reduce security risks and fraud, bringing transparency to an unprecedented scale through its immutability, security, and transparency in a permissionless and decentralized environment [2]. As of July 2024, Ether, the cryptocurrency of the Ethereum network, has a market capitalization exceeding $370 billion, with millions of transactions executed [3]. However, the same features that make Ethereum powerful also attract attackers seeking to exploit vulnerabilities within smart contracts [4]. Recent surveys have shown that smart contracts are highly susceptible to attacks such as reentrancy, arithmetic overflows/underflows, front-running, and others, leading to significant financial losses [4][5]. For instance, the Level Finance Exchange experienced a loss of more than $1.01 million due to a recursive calling vulnerability in May 2023 [4].

As the basic architecture, Ethereum blockchain operates as a Layer 1 (L1) solution, providing the fundamental base layer for the network’s security, consensus, and transaction processing. However, L1 blockchains like Ethereum face limitations in scalability and efficiency due to the need to process every transaction and store all data on-chain, which leads to congestion and slower transaction times during periods of high demand. To address these challenges, the proposed solution considers the InterPlanetary File System (IPFS), used to offload storage and computational tasks from the main blockchain. The IPFS solutions enhance scalability and performance by handling data storage and other operations off-chain, thus reducing the load on the L1 blockchain.

The InterPlanetary File System (IPFS) is a decentralized content-addressable object storage and retrieval platform that is part of the Decentralized Web effort. IPFS allows computers worldwide to store and serve files as part of a giant peer-to-peer network [6]. Any computer can download the IPFS software and start hosting and serving files, making it a decentralized solution for data storage [6]. Files added to IPFS are given a unique address derived from a hash of the file's content, known as a Content Identifier (CID), ensuring the integrity and immutability of the stored data [6]. IPFS uses a Distributed Hash Table (DHT) to map from CIDs to nodes storing the content, enhancing the robustness and availability of the data [6]. When content is uploaded to IPFS, it is announced to the network, and the data can be retrieved by looking up the CID in the DHT [6]. This decentralized nature of IPFS distributes the storage and retrieval responsibilities across the network, improving the system's performance and reliability [7].

Additionally, IPFS supports the storage of directories and entire static websites, where each file and folder receive its own CID [6]. This feature allows for efficient management and retrieval of related files, enhancing the versatility of the IPFS system [6]. The adoption of IPFS, however, faces challenges due to its performance being slower than traditional HTTP for content retrieval and for the requirement to run IPFS node software by users, which demands specific technical skills and resources [7].

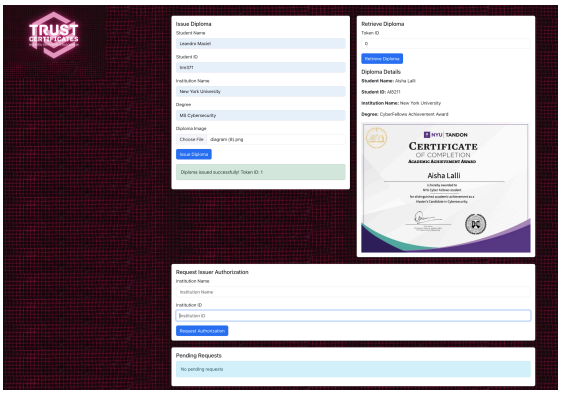

Figure 1. Front-End https://github.com/allali7/diploma-nft

Furthermore, there are concerns regarding the use of third-party IPFS providers, which can introduce vulnerabilities if the content they store is deleted or becomes inaccessible. This issue highlights the need to address and manage such vulnerabilities to ensure true immutability and permanence of the stored data.

Despite these challenges, IPFS offers a viable solution for decentralized storage, reducing the concentration of control and enhancing data security and accessibility [6]. By leveraging IPFS for decentralized storage and Ethereum for immutable record keeping, the "Immortal Certificates" system aims to provide a secure, permanent, and reliable solution for digital certificate management.

3. Purpose and Necessity of the "Immortal Certificates" Project

3.1. Addressing the Gaps in Current Digital Certificate Systems

In the digital age, the issuance and verification of credentials such as academic degrees, professional certifications, and legal documents have become increasingly crucial. However, traditional digital certificate systems are plagued by several significant shortcomings:

3.1.1. Centralization Risks: Most existing digital certificate systems rely on centralized authorities, which act as the sole issuers and verifiers of certificates. This centralization introduces a single point of failure, making the entire system vulnerable to tampering, data breaches, and the risk of loss. If the central authority is compromised or ceases to operate, the integrity and accessibility of all certificates under its control are jeopardized.

3.1.2. Forgery and Fraud: Traditional systems' centralized nature makes them susceptible to forgery and fraud. Malicious actors can manipulate these systems to create counterfeit certificates that are difficult to detect. This not only undermines the trust in digital credentials but also poses a significant risk to organizations and individuals who rely on these certificates for verification.

3.1.3. Lack of Perpetuity: Certificates issued through traditional systems may not be accessible or verifiable in the long term. Technological changes, organizational restructuring, or even the dissolution of issuing authorities can render certificates inaccessible or unverifiable, leading to potential legal and professional complications.

3.2. The Unique Value of "Immortal Certificates"

The "Immortal Certificates" project is designed to address these gaps by leveraging the decentralized and immutable nature of blockchain technology. The system provides a robust solution that ensures the perpetual existence, security, and verifiability of digital certificates. Here's how the project stands out:

3.2.1. Immutability and Integrity: By issuing certificates as Non-Fungible Tokens (NFTs) on the Ethereum blockchain, the "Immortal Certificates" system guarantees that once a certificate is issued, it cannot be altered or tampered with. This immutability ensures the long-term integrity and trustworthiness of digital credentials, making them resistant to fraud and forgery.

3.2.2. Decentralization and Resilience: The use of blockchain technology eliminates the reliance on a

single authority, distributing the control and verification of certificates across a decentralized network. This decentralization enhances the resilience of the system, ensuring that certificates remain accessible and verifiable even if individual nodes or entities are compromised or go offline.

3.2.3. Perpetual Availability: The "Immortal Certificates" project ensures that certificates are stored and maintained indefinitely. By utilizing decentralized storage solutions such as the InterPlanetary File System (IPFS), the system guarantees that certificates can be accessed and verified at any point in the future, regardless of changes in technology or organizational structures.

3.3. The Unique Value of "Immortal Certificates"

A key differentiator of the "Immortal Certificates" project is its foundation in cybersecurity. Unlike many existing solutions that focus primarily on the technological aspects of blockchain, this project incorporates advanced cybersecurity measures from the outset. This cybersecurity-focused approach includes:

3.3.1. Security-by-Design Architecture: The system is built with security as a core principle, incorporating best practices in encryption, hashing, and secure coding to prevent vulnerabilities and ensure the protection of sensitive data.

3.3.2. Preparation for Future Threats: As quantum computing technology advances, it poses a potential threat to current cryptographic algorithms. The "Immortal Certificates" project will align with the National Institute of Standards and Technology (NIST) recommendations by integrating quantum-resistant encryption algorithms. This proactive approach ensures that the system remains secure even as new technological threats emerge.

3.3.3. Unique Perspective: The project benefits from the expertise of cybersecurity professionals, who bring a deep understanding of potential vulnerabilities and attack vectors. This knowledge shapes the design and implementation of the system, resulting in a more robust and resilient solution than what is currently available in the market.

As the world becomes increasingly digital, the need for secure, reliable, and perpetual verification of credentials will only grow. The "Immortal Certificates" project offers a forward-thinking, comprehensive solution that addresses the critical shortcomings of traditional digital certificate systems. By combining blockchain technology with advanced cybersecurity practices and preparing for future technological developments, this project ensures the secure, immutable, and perpetual management of digital credentials.

4. Related Work

Digital certificate systems have evolved significantly, with various solutions attempting to address the inherent limitations of traditional centralized systems. These centralized systems, which rely on a single authority for issuing and verifying certificates, are vulnerable to tampering, loss, unauthorized access, and forgery. Furthermore, they often involve cumbersome and costly processes for institutions.

DiplomaSafe, for example, is a platform that digitizes the issuance of diplomas, providing a streamlined process for educational institutions while aiming to ensure the trustworthiness of digital diplomas in a centralized architecture [8]. Currently there are studies to create a private blockchain solution for DiplomaSafe, even considering its overhead costs. By leveraging blockchain technology for timestamping and ensuring the provenance of diplomas, DiplomaSafe enhances the security and authenticity of these credentials. Despite these benefits, the adoption of blockchain technology introduces challenges related to the technical skills required and the integration with existing systems [9],[10].

Blockchain-based solutions offer a promising alternative, leveraging the decentralized and immutable nature of blockchain to issue certificates as Non-Fungible Tokens (NFTs). These systems provide enhanced security, transparency, and permanence. However, scalability remains a significant concern, as the transaction throughput of blockchain networks like Ethereum can be limited during peak times. Additionally, the cost of transactions can fluctuate, affecting the feasibility of widespread adoption.

Decentralized storage solutions such as the InterPlanetary File System (IPFS) address some of these scalability issues by offloading data storage from the blockchain to a distributed network. IPFS ensures data integrity and accessibility by using content-addressable storage, which assigns a unique hash to each file. This approach reduces the load on the blockchain but introduces challenges related to data availability and retrieval speed. The performance of IPFS can be slower than traditional centralized storage methods, and as mentioned earlier there are concerns about the reliability of third-party IPFS providers [11], [12], [13].

This current research aims to address the gaps in existing solutions by proposing a system that ensures the true immortality and security of certificates through a decentralized, blockchain-based solution. By utilizing Ethereum for immutable record-keeping and IPFS for decentralized storage, we aim to create a scalable and secure system for digital certificate management. The process of refining and enhancing this system is in its early stages, building on the lessons learned from existing solutions and addressing their limitations.

5. System Design and Implementation

The system architecture is designed to ensure the perpetual existence, security, and verifiability of digital certificates by combining several key technologies. The primary components of the system include the Ethereum blockchain for decentralized ledger storage, smart contracts for managing the issuance and verification of certificates, and the InterPlanetary File System (IPFS) for off-chain data storage.

5.1. Ethereum Blockchain for Decentralized Ledger Storage

The Ethereum blockchain serves as the system's foundational layer, providing a decentralized ledger that records all transactions related to the issuance and verification of certificates. This ensures that the data is tamper-proof, transparent, and accessible to all participants. Each certificate issued is represented as a Non-Fungible Token (NFT) on the blockchain, guaranteeing its uniqueness and immutability. The blockchain's decentralized nature ensures that no single entity has control over the entire system, reducing the risk of fraud and increasing trust in the certificates issued.

5.2. Smart Contracts for Issuance and Verification

Smart contracts are self-executing contracts with the terms of the agreement directly written into code. In Immortal Certificates, smart contracts manage the entire lifecycle of a certificate, from issuance to verification. The smart contract logic includes several critical functions: The Ethereum blockchain is foundational for several purposes.

5.2.1. Issuer Authorization: Institutions that wish to issue certificates must first be authorized by the system. This process involves submitting a request through the smart contract, which is then subject to approval by existing authorized issuers. This ensures that only legitimate institutions can issue certificates.

5.2.2. Voting Mechanisms: To prevent unauthorized issuance, the system includes a voting mechanism where existing authorized issuers must approve new requests for issuer authorization. The voting process is decentralized, with each authorized issuer having a vote. This decentralized voting mechanism ensures that no single institution can dominate the decision-making process.

5.2.3. Certificate Issuance: Once an issuer is authorized, they can issue certificates as NFTs. Each certificate is linked to an IPFS hash that points to the actual certificate data stored off-chain. This ensures that the certificate's content remains immutable and verifiable.

5.2.4. Certificate Verification: The smart contract also includes functions for verifying the authenticity of certificates. Users can query the blockchain to retrieve certificate details and verify their authenticity against the immutable records stored on the blockchain.

5.3. IPFS for Decentralized Data Storage with Pinata

While the Ethereum blockchain provides a secure and immutable ledger, it is not suitable for storing large amounts of data due to scalability and cost constraints. To address this, the system uses the InterPlanetary File System (IPFS) for off-chain storage of certificate data. IPFS is a decentralized storage solution that allows for storing and retrieving files in a distributed manner. For improved performance and reliability, this system uses Pinata, a third-party IPFS service, to manage and pin the files on the IPFS network. Pinata ensures that the files remain accessible and are stored reliably across the IPFS network.

5.3.1. Content Addressing with CIDs: Each file stored in IPFS via Pinata is assigned a unique Content Identifier (CID) based on the file's content. This CID acts as a fingerprint for the file, ensuring its integrity. If the content of the file changes, the CID will change, making it impossible to tamper with the data without detection.

5.3.2. Redundancy and Resilience: IPFS operates as a peer-to peer (P2P) network, where files are broken into smaller chunks and distributed across multiple nodes. Pinata ensures that these files are pinned, meaning that they are stored persistently on various nodes. This distribution ensures redundancy, meaning that even if some nodes go offline, the data can still be retrieved from other nodes hosting the same content. This resilience is crucial for ensuring the long-term availability of certificate data.

5.3.3. Decentralized Storage Benefits: With Pinata's support, certificate data can be stored off-chain in IPFS, reducing the load on the Ethereum blockchain and minimizing transaction costs. Additionally, IPFS's decentralized nature aligns with the goal of eliminating single points of failure, further enhancing the system's security and reliability.

5.4. Incentive Mechanisms for Legitimate Participation

To ensure that only legitimate institutions participate in the issuance process, the system incorporates an incentive mechanism within the smart contracts. Institutions that wish to issue certificates must pay a fee, which is then distributed among the participants in the voting process. This fee serves multiple purposes:

5.4.1. Encouraging Participation: By distributing fees among voters, authorized institutions are

incentivized to actively participate in the decision-making process, ensuring that only legitimate institutions are authorized to issue certificates.

5.4.2. Maintaining System Integrity: The fee structure also discourages malicious actors from attempting to flood the system with unauthorized requests, as each request requires a financial commitment.

5.4.3. Sustaining the Network: The fees collected help sustain the network by compensating those who store and maintain the certificate data, ensuring the long-term viability of the system.

5.5. Achieving True Immutability and Perpetuity

The combination of Ethereum's blockchain for immutable record-keeping and IPFS (with Pinata) for decentralized storage creates a robust system that addresses the limitations of traditional digital certificate systems. By leveraging the strengths of both technologies, certificates issued through the system are not only secure and verifiable but also perpetual in their existence. This approach provides a level of trust and reliability that is essential for the digital credentials of the future.

6. Smart Contract Logic

The smart contract logic for our system is encapsulated in the DiplomaNFT contract, which leverages the Ethereum blockchain for decentralized ledger storage and IPFS for off-chain storage of diploma data. The contract includes several vital functions and mechanisms to manage the issuance and verification of diplomas, as well as the authorization of issuers. The following flowcharts provide a visual representation of the core processes in the contract:

6.1. Key Components and Functions

6.1.1. Issuer Authorization: The contract includes a mechanism for institutions to request authorization to issue diplomas. An issuer submits a request along with a fee, and existing authorized issuers vote on whether to approve the new issuer. This process ensures that only trusted institutions can issue diplomas.

6.1.2. Voting Mechanism: Authorized issuers can vote on new issuer requests. If the request receives the required number of approvals, the new issuer is authorized. If rejected, the request is marked as processed.

```
function requestAuthorization(string memory
    institutionName, string memory institutionID)
    public payable limitPendingRequests {
    require(msg.value >= REQUEST_FEE, "Insufficient
        fee");
    require(!authorizedIssuers[msg.sender].isActive,
        "Already an authorized issuer");
    require(block.timestamp > authorizedIssuers[msg.
        sender].lastVotedTimestamp +
        REQUEST_COOLDOWN, "Cooldown period active");

    bytes32 requestId = keccak256(abi.encodePacked(
        msg.sender, block.timestamp));
    IssuerRequest storage request = issuerRequests[
        requestId];
    request.requester = msg.sender;
    request.institutionName = institutionName;
    request.institutionID = institutionID;
    pendingRequests++;
    pendingRequestIds.push(requestId);

    emit IssuerRequestSubmitted(requestId, msg.
        sender, institutionName, institutionID);
}
```

Figure 2. Function for Issuer Authorization.

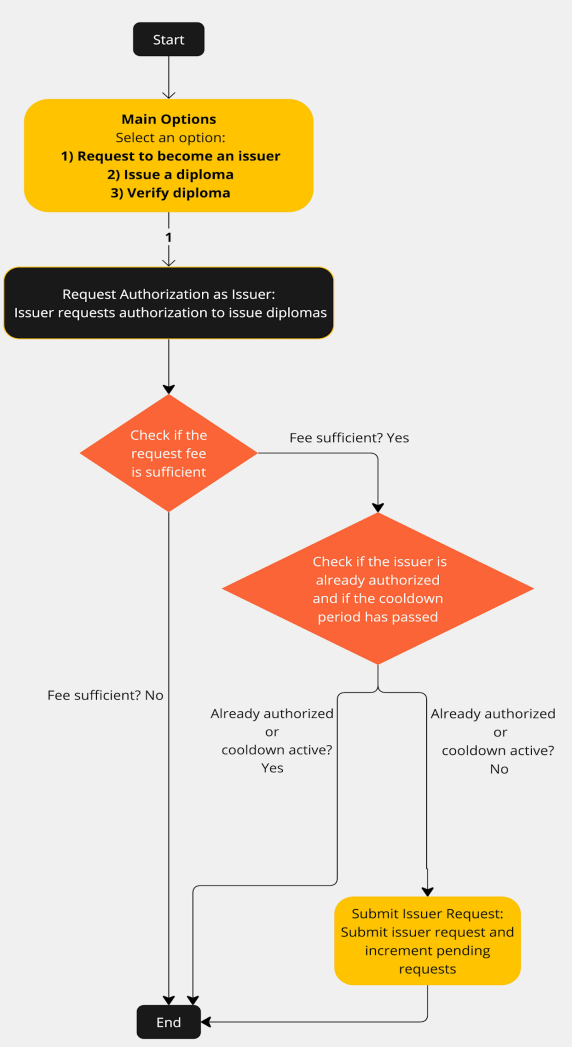

Figure 3. Issuer Authorization Request Flowchart. This flowchart outlines the process for an issuer to request authorization to issue diplomas.

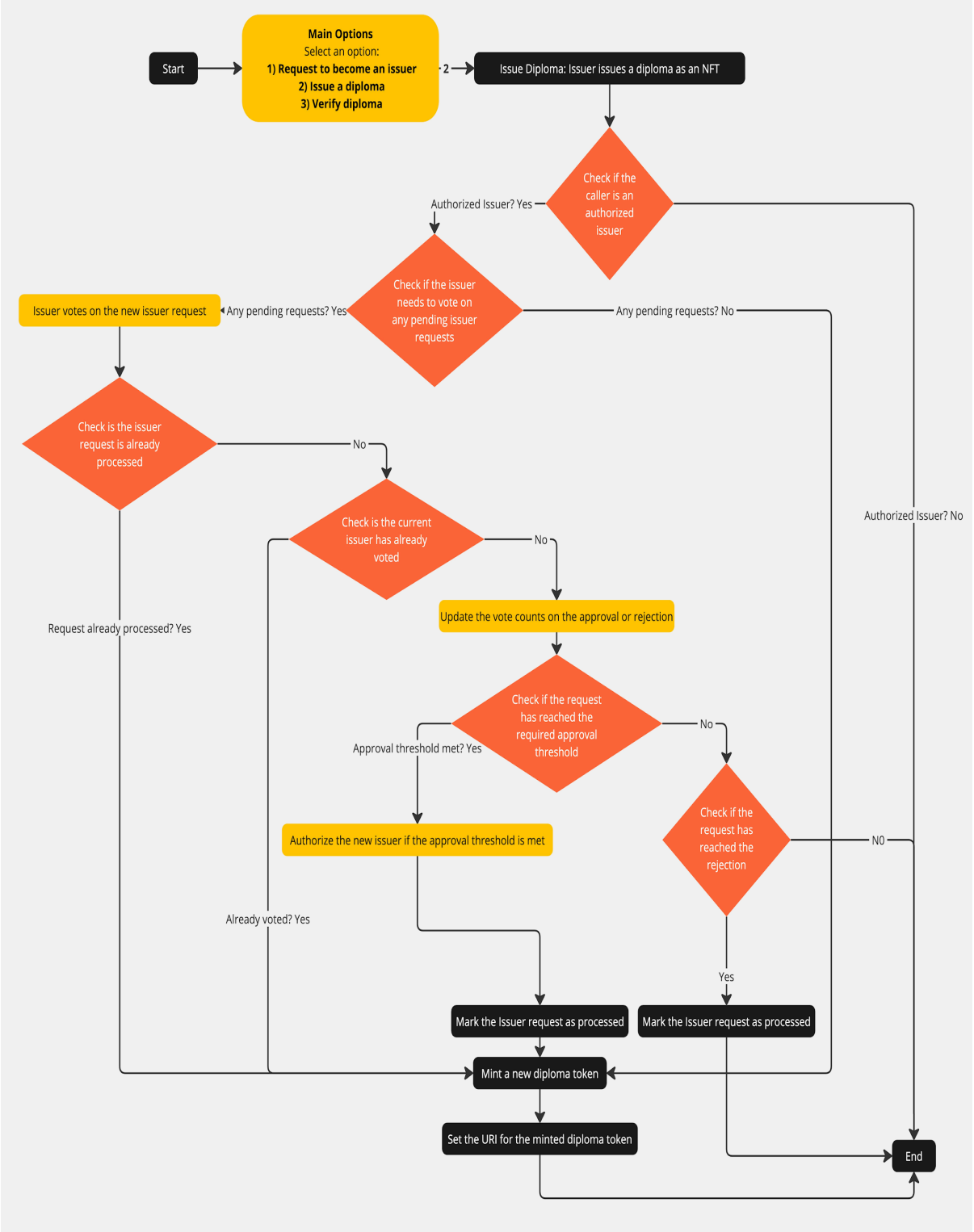

Figure 4. Diploma Issuance and Issuer Voting Flowchart. This flowchart describes the process for issuing a diploma and the voting mechanism for approving new issuer requests.

```
function voteOnIssuerRequest(bytes32 requestId, bool
     approve) public onlyAuthorizedIssuer {
    IssuerRequest storage request = issuerRequests[
        requestId];
    require(!request.processed, "Request already
        processed");
    require(!request.voted[msg.sender], "Already
        voted");

    request.voted[msg.sender] = true;
    authorizedIssuers[msg.sender].lastVotedTimestamp
         = block.timestamp;

    if (approve) {
        request.approvals++;
    } else {
        request.rejections++;
    }

    emit IssuerRequestVoted(requestId, msg.sender,
        approve);

    uint256 issuerCount = getIssuerCount();
    uint256 requiredApprovals = (issuerCount * 65) /
         100;

    if (request.approvals >= requiredApprovals) {
        authorizedIssuers[request.requester] =
            Issuer(true, request.institutionName,
            request.institutionID, block.timestamp,
            block.timestamp, 0);
        request.processed = true;
        issuers[request.requester] = true;
        pendingRequests--;
        emit IssuerAuthorized(request.requester);
    } else if (request.rejections > (issuerCount /
        2)) {
        request.processed = true;
        pendingRequests--;
    }
}
```

Figure 5. Function with Voting Mechanism.

6.1.3. Diploma Issuance: Authorized issuers can issue diplomas as NFTs, storing essential details such as the student's name, institution, and an IPFS hash of the diploma document. The process ensures each diploma is unique and immutable.

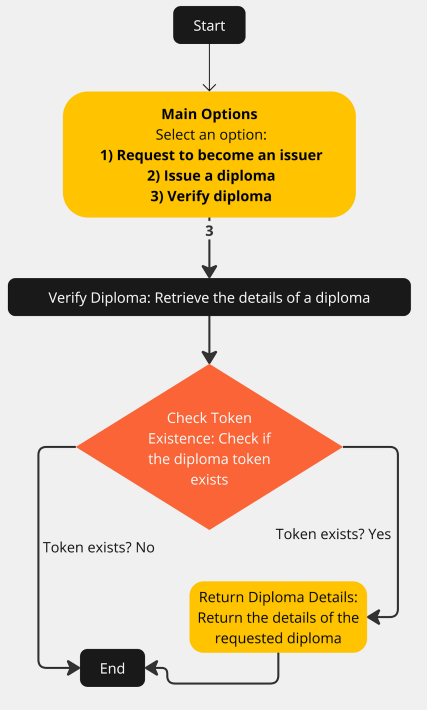

Figure 6. Diploma Verification Flowchart. This flowchart illustrates the process for verifying a diploma.

```
function issueDiploma(
    string memory studentName,
    string memory studentID,
    string memory institutionName,
    string memory degree,
    string memory ipfsHash,
    string memory _tokenURI
) public onlyAuthorizedIssuer returns (uint256
    tokenId) {
    if (pendingRequests > 0 && authorizedIssuers[msg
        .sender].diplomasIssuedSinceLastVote >=
        DIPLOMAS_BEFORE_VOTING) {
        bytes32 firstRequestId =
            getFirstPendingRequest();
        voteOnIssuerRequest(firstRequestId, true);
        authorizedIssuers[msg.sender].
            diplomasIssuedSinceLastVote = 0; //
            Reset the count after voting
    }

    tokenId = _tokenIds.current();
    _tokenIds.increment();

    _safeMint(msg.sender, tokenId);
    _setTokenURI(tokenId, _tokenURI);

    // Store additional diploma data
    diplomaToStudentID[tokenId] = studentID;
    diplomaToInstitutionName[tokenId] =
        institutionName;
    diplomaToIpfsHash[tokenId] = ipfsHash;
    diplomaToStudentName[tokenId] = studentName;

    authorizedIssuers[msg.sender].
        diplomasIssuedSinceLastVote++;

    emit DiplomaIssued(tokenId, msg.sender,
        studentName, degree);
}
\subsubsection{Diploma Verification}
Users can verify the authenticity of a diploma by
    retrieving the details of the diploma NFT. The
    system checks if the diploma token exists and
    returns the corresponding metadata.

function getDiploma(uint256 tokenId) public view
    returns (string memory) {
    require(ownerOf(tokenId) != address(0), "Diploma
         not found");
    return tokenURI(tokenId);
}
```

Figure 7. Diploma Issuing and Verification Function.

7. Incentivizing Voter Participation

Initially, the system forced institutions to vote before issuing diplomas. Specifically, the contract required authorized issuers to participate in the voting process for new issuer requests before they could issue any diploma. This was implemented to ensure that the authorization of new issuers remained a collective process. However, this approach proved inefficient as it created bottlenecks and delays, particularly when issuers had pending requests and needed to issue diplomas urgently.

To address this issue, a new incentive mechanism was introduced, where requestors pay a fee for authorization, which is then distributed among voters who participate in the decision-making process. This mechanism serves multiple purposes:

7.1. Incentivizing Participation:

By distributing the authorization fee among voters, an incentive for institutions to actively participate in the voting process is created.

7.2. Ensuring Fairness:

The fee structure and distribution are designed to be equitable, ensuring that all participating voters receive a fair share of the fees, promoting a balanced and decentralized decision-making process.

7.3. Timely Decision-Making:

The system includes required voter participation thresholds to prevent delays. These thresholds ensure that decisions are made promptly without causing unnecessary delays in the issuance of diplomas.

The initial contract logic included a direct enforcement mechanism for voting:

```
if (pendingRequests > 0 && authorizedIssuers[msg.
    sender].diplomasIssuedSinceLastVote >=
    DIPLOMAS_BEFORE_VOTING) {
    bytes32 firstRequestId = getFirstPendingRequest
        ();
    voteOnIssuerRequest(firstRequestId, true);
    authorizedIssuers[msg.sender].
        diplomasIssuedSinceLastVote = 0; // Reset
        the count after voting
}
```

Figure 8. Enforcement Mechanism for Voting.

This segment of the contract forced issuers to vote on at least one pending request before they could continue issuing more diplomas, which often led to inefficiencies.

By shifting to the incentive-based model, operational delays were reduced, and a more dynamic and responsive voting process was created. The new approach balances incentives and ensures timely decision-making, ultimately enhancing the system's efficiency and reliability.

8. Security and Scalability

Security is a paramount concern in the proposed system. By using IPFS for decentralized storage, the risks associated with centralized data repositories are mitigated. However, third-party IPFS providers, while useful, can still be susceptible to data loss. To achieve true immortality, issuers were incentivized to store and maintain redundant full IPFS data also through a fee distribution mechanism. Additionally, encryption, hashing, and salting techniques were employed to protect certificate data from unauthorized access and tampering. Scalability is addressed by optimizing smart contract functions and ensuring efficient data handling to support large numbers of certificates and requests.

8.1. Current Security Issues with Digital Certificates

Digital certificates in traditional systems are often stored in centralized databases, making them vulnerable to tampering, loss, and unauthorized access. Centralized authorities control ling these certificates can be single points of failure, and the certificates themselves can be revoked or altered without the knowledge of the certificate holders.

8.2. Blockchain Technology as a Solution

Blockchain technology, with its decentralized and immutable ledger, offers a robust solution to these issues. By issuing certificates as Non-Fungible Tokens (NFTs) on the Ethereum blockchain, each certificate is unique, tamper-proof, and permanently recorded on the blockchain. This immutability is crucial for maintaining the integrity and trustworthiness of the certificates.

8.3. Decentralized Storage with IPFS

While blockchain ensures the integrity and immutability of the certificates, it is not suitable for storing large amounts of data. This is where the InterPlanetary File System (IPFS) comes in. IPFS is a decentralized storage solution that allows us to store the actual certificate data off-chain, while the blockchain stores only the IPFS hash, ensuring the certificate's content remains immutable and easily retrievable.

Despite its advantages, IPFS faces significant challenges, including increased complexity and overhead, performance compromises, and scalability issues. For example, the decentralized routing process of node discovery in the distributed hash table (DHT) can significantly hinder the publication of large amounts of content, leading to delays and high traffic volumes [14]. Content retrieval performance in IPFS is approximately three times slower than HTTPS, making it less suitable for real-time applications such as live video streaming. This is due to the decentralized nature of IPFS, which requires several routing hops in the DHT to locate the content [14]. Adoption of IPFS is hindered by the need for users to run IPFS node software, which requires specific technical skills and cannot yet run on mobile devices due to its footprint. Additionally, integrating IPFS with traditional HTML/HTTP websites is challenging because IPFS retrievals tend to be slower than HTTP [14]. While decentralization enhances security and privacy by reducing the vantage of individual nodes, centralized components introduce trade-offs that need to be carefully managed. For example, centralized gateways can aggregate demand and enable caching, improving performance but also introducing potential points of failure and control [14].

8.4. Limitations of Third-Party IPFS Providers

Using third-party IPFS providers has its limitations. These providers, while offering a gateway to IPFS, remain centralized services and can be subject to failures or data loss. If a certificate

stored on a third-party IPFS provider is deleted, it is reflected on the network, and the certificate data gets lost [14]. This vulnerability undermines the immutability promise. To achieve true immortality, full nodes in the network are incentivized to store and maintain the data. This can be done by distributing the fee paid by requestors among the voters who participate in the authorization process [14].

8.5. Incentive Mechanism for Full IPFS Backup

To encourage active participation in the voting process and ensure data longevity, the following incentive mechanism is proposed. Requestors who want to become authorized issuers will pay a fee, which will be distributed among the voters. Also, some monetary distribution will be given to universities that participate in network assurance, keeping full backups of IPFS data and ensuring the data is continuously stored and maintained by multiple stakeholders, thereby achieving true decentralization and immortality.

8.6. Security Measures

8.6.1. Encryption: Encryption techniques were used to secure the data stored on IPFS and protect the confidentiality of certificate data.

8.6.2. Hashing: Hash functions are used to generate a unique hash for each certificate, ensuring data integrity and preventing tampering.

8.6.3. Salting: Adding salt to the hashes ensures that even if two certificates have similar content, their hashes will be unique, further enhancing security.

8.7. Secure by Design Practices in DiplomaNFT Smart Contract

The security of a smart contract is paramount, especially when dealing with sensitive data such as digital certificates. The DiplomaNFT smart contract is designed with multiple security practices to ensure its robustness against various threats. The key secure-by-design practices implemented in the smart contract are presented below.

8.7.1. Common Vulnerabilities:

• **Reentrancy**: This occurs when a contract calls another contract, allowing for recursive calls, potentially leading to unintended consequences. Mitigation involves using the checks-effects-interactions pattern and mutexes [15].

• **Arithmetic Issues**: These include integer overflows and underflows. The use of Solidity's SafeMath library helps mitigate these vulnerabilities by ensuring arithmetic operations revert on overflow [15].

• **Front-running**: This happens when miners or participants can manipulate transaction order to their advantage. Implementing commit-reveal schemes can help mitigate this risk [15].

• **Mishandled Exceptions**: These arise when low-level calls fail silently. Proper return value handling and high-level calls can prevent such issues [15].

• **Code Injection via *delegatecall***: This vulnerability allows malicious code execution within the context of the calling contract. Assigning delegate calls to trusted contracts and using stateless library contracts can mitigate this risk [15].

• **Randomness Using Block Information**: Block information-based randomness is susceptible to miner manipulation. Using oracles or cryptographic commitment schemes randomness [15].

8.7.2. Tools and Methodologies for Vulnerability Detection: Various tools have been developed to detect and mitigate vulnerabilities in smart contracts. The most prominent ones include:

• **Oyente**: Uses symbolic execution to detect vulnerabilities such as reentrancy and transaction order dependence [15].

• **Slither** A static analysis tool that can detect a wide range of issues, including reentrancy and code injection [15].

• **Mythril**: Performs symbolic execution and can uncover various vulnerabilities by generating concrete exploit values [15].

• **Manticore**: Uses symbolic execution for in-depth analysis and can identify vulnerabilities at the EVM bytecode level [15].

• **Echidna**: A fuzzing tool that tests smart contract properties to uncover potential issues [15].

Combining these tools can provide comprehensive coverage and help ensure the contract's security. While each tool has its strengths and limitations, their combined use can mitigate false positives and provide deeper insights into potential vulnerabilities [15].

8.7.3. Principle of Least Privilege: Ensure that each part of the contract has the minimum privileges necessary to perform its function. This minimizes the potential impact of any vulnerabilities.

• Use of modifiers like *onlyAuthorizedIssuer* to restrict access to sensitive functions.

• Ensure that only authorized issuers can issue diplomas or vote on issuer requests.

8.7.4. Input Validation: Validate all inputs to prevent unexpected behavior and vulnerabilities like integer overflow/underflow or invalid data being processed.

• Validate all inputs in critical functions, such as requestAuthorization and voteOnIssuerRequest, to ensure they meet the expected formats and conditions.

• Use *require* statements to enforce conditions before processing.

8.7.5. Safe Math Operations: Use safe math operations to prevent overflow and underflow vulnerabilities.
• Utilize the SafeMath library from OpenZeppelin to handle arithmetic operations securely.
8.7.6. Immutable Data Storage: Ensure that critical data, once written, cannot be altered to preserve the integrity of the certificates.
• Use *constant* and *immutable* keywords for variables that should not change after contract deployment.
8.7.7. Decentralized and Transparent Governance: Implement a decentralized governance model to avoid central points of failure and enhance trust.
• Use a voting mechanism where authorized issuers participate in decision-making.
• Distribute the decision-making process across multiple participants to ensure fairness and transparency.
8.7.8. Event Logging for Transparency and Debugging: Emit events for critical operations to provide transparency and facilitate debugging.
• Emit events for actions such as issuing diplomas, submitting issuer requests, and voting on requests.
8.7.9. Security Measures for Off-Chain Storage: Ensure that off-chain storage is secure and resilient.
• Use IPFS for decentralized storage and ensure data integrity through hashing.
• Address limitations of services like third-party IPFS providers by incentivizing full nodes to store data.
8.7.10. Regular Audits and Testing: Conduct regular security audits and thorough testing to identify and fix vulnerabilities.
• Regularly audit the smart contract code by independent security experts.
• Use automated testing frameworks to simulate various scenarios and edge cases.
8.7.11. Using Proven Libraries and Frameworks: Leverage well-established libraries and frameworks to minimize the risk of introducing vulnerabilities.
• Utilize OpenZeppelin's contracts for ERC721 implementation, safe math operations, and access control.

8.8. Mitigating DDoS Attacks

To protect the smart contract from Distributed Denial of Service (DDoS) attacks, given the front-end access, the implementation of several strategies was proposed:
8.8.1. Fee for Authorization Requests: By charging a fee for requesting to become an authorized user, frivolous or malicious requests could be deterred, reducing the risk of DDoS attacks on the authorization process [15].
8.8.2. Robust Wallet Registration: Requiring wallet registration adds a layer of security. However, additional measures must be considered to prevent DDoS attacks on the token validation process [15].
8.8.3. Voting Universities and Incentives: Utilizing authorized universities for voting ensures that a network of trusted participants verifies any changes or validations. Universities or issuers can receive payments for their participation, incentivizing their involvement and ensuring the system remains decentralized and secure [15].

To further secure the system against DDoS attacks, especially on the receive validate token part, several measures are implemented:
• **Fee-Based Throttling:** Implement a fee for issuing a request to be an authorized user and validating tokens to deter bulk automated requests.
• **Rate Limiting:** Restrict the number of requests an address can make within a certain period.
• **Proof of Work (PoW) Challenges**: Requiring requesters to solve computationally simple puzzles increases the cost of performing large-scale attacks.
• **CAPTCHA Integration:** Use CAPTCHA to prevent automated bots from flooding the system.
• **Multi-Layered Authentication:** Combine wallet registration with additional authentication layers, such as two factor authentication (2FA).
• **IP Whitelisting and Blacklisting:** Maintain and dynamically update lists of trusted and malicious IP addresses.
• **Monitoring and Alerts:** Implement real-time monitoring and alerting systems to detect and respond to unusual activity patterns.
• **Decentralized Verification Nodes:** Distribute verification across multiple nodes to avoid single points of failure.
• **Escalating Fees:** Increase fees for repeated requests from the same address within a short timeframe.
• **Timeouts and Expirations:** Discard requests not processed within a specific timeframe to prevent abuse.
• **Community Voting for Blocking Abusers:** Allow the community to vote on blocking abusive users or addresses.

9. Evaluation and Results

This chapter presents the evaluation and results of an Ethereum-based Blockchain application designed for the decentralized and immutable storage of university diplomas. The system, developed based on the principles and considerations discussed earlier, was implemented and assessed across various performance metrics, including response time, cost, and security. The experimental setup involved deploying the smart contract on the Ethereum blockchain. The critical components utilized were:
• **Solidity:** For writing the Smart Contract;
• **REMIX IDE:** For editing, compiling, and deploying the Smart Contract;

• **MetaMask:** As the Ethereum Wallet;
• **Sepolia:** As the Ethereum Testnet;
• **Etherscan:** For validating the contract creation; and
• **Pinata:** For decentralized storage via IPFS.

9.1. Overview of System Functions

The system's user interface provides three primary functions:

9.1.1. Issuing a Diploma: by an authorized entity such as a university.

9.1.2. Retrieving a Diploma: available to anyone with the appropriate Token ID.

9.1.3. Requesting Issuer Authorization: where institutions apply to become authorized to issue diplomas.

The platform is built on the Ethereum Blockchain and leverages smart contracts executed in an Ethereum Virtual Machine (EVM). Transactions within the system require "gas," a small payment in Ethers to cover the cost of processing operations on the Ethereum Blockchain. This test was conducted on the Sepolia Testnet, simulating real-world usage without incurring significant transaction fees.

9.2. Requesting Issuer Authorization

The first function tested was the request for authorization to issue diplomas. The process begins with the institution entering its official name and an identification number or nickname, known as the "Institution ID." This information and the transaction are submitted on the Ethereum Blockchain, costing approximately 0.002 Ethers in gas fees. The transaction awaits approval in a decentralized manner, utilizing a Proof of Stake (PoS) model. Existing universities within the system validate the request, adding a layer of security and decentralization to the approval process.

9.3. Issuing and Retrieving Diplomas

Next, the "Issue Diploma" function was evaluated. The interface presents several customizable fields where the institution enters relevant student information. Accuracy is important, as these fields are stored on the blockchain as immutable metadata. To minimize storage costs and avoid overloading the blockchain, the actual diploma image (a PNG file) is stored off-chain using a decentralized storage solution, the InterPlanetary File System (IPFS).

Issuing the diploma required 0.009 Ethers, reflecting the cost of permanently storing data on the blockchain. Note that this cost varies, depending on the time of the day the operation was executed. After successfully issuing the diploma, a unique Token ID is assigned, which acts as the diploma's identifier on the blockchain. In this trial, Token ID 0 was assigned to the first diploma.

The retrieval process was then tested. Using the Retrieve Diploma function, Token ID 0 was successfully retrieved, confirming the accuracy of the stored diploma and associated metadata. The retrieval process does not require gas, making it cost-effective for users who want to verify diplomas.

During the test run, the system could store and retrieve multiple diplomas without issues. However, a minor error occurred when the same diploma was issued twice, creating two identical tokens with IDs 0 and 1. This redundancy highlights a potential improvement area for the system, where mechanisms could be developed to check for duplicate tokens and prevent unnecessary storage. Token ID 2 was successfully issued for a different student, and the system confirmed the proper storage of this token as well.

Diploma retrieval was also tested on a different machine with a separate Ethereum account and wallet, demonstrating that the system is fully decentralized. No gas fees were required to retrieve tokens, and diplomas with Token IDs 0, 1, and 2 were successfully validated across different devices [16].

9.4. Future Improvements

While the system functions effectively, there are opportunities for refinement. For example, a verification feature to detect redundant diploma entries could optimize storage use (e.g., salting techniques). Expanding the system’s capabilities to track multiple diplomas for the same student with distinct metadata would further enhance its usability.

Also, although this test run demonstrated that the Ethereum Blockchain, combined with IPFS, can provide a decentralized, transparent, and secure system for issuing and verifying academic credentials, while functional, the system can be improved to optimize storage and validation processes. Future work may involve integrating additional smart contract features to prevent errors and ensure seamless operation across diverse scenarios [17].

10. Conclusion

The "Immortal Certificates" project provides a solution for digital certificate issuance and management, addressing the vulnerabilities of traditional centralized systems. Utilizing the Ethereum blockchain and the InterPlanetary File System (IPFS) ensures the security, immutability, and perpetual availability of academic and professional credentials. The blockchain offers an immutable ledger that guarantees the certificates' integrity, while IPFS handles off-chain storage, reducing the on-chain data load and ensuring scalability and cost-effectiveness.

A key strength of this system lies in its security-by-design architecture, which integrates cybersecurity principles from the outset. This approach ensures that potential security vulnerabilities are minimized and the system is resilient to emerging threats. The project incorporates industry best practices such as encryption, hashing, and salting to protect sensitive data. Encrypting certificate data stored off-chain and generating unique hashes for each certificate guarantees data integrity and prevents tampering. Salting further enhances security by ensuring that even if two certificates have similar content, their digital fingerprints remain unique.

The project also addresses the growing threat of quantum computing, which could potentially undermine current cryptographic algorithms. By aligning with the National Institute of Standards and Technology (NIST) guidelines, the system is designed to incorporate quantum-resistant encryption algorithms, ensuring long-term security even as quantum computing technology advances.

One of the primary innovations of the "Immortal Certificates" system is its decentralized governance model. The authorization of new certificate issuers is managed through a decentralized voting mechanism based on a Proof of Stake (PoS) model. This ensures that no single entity controls the issuance process, mitigating the risks associated with central points of failure. Additionally, the incentive mechanism for voter participation helps maintain system integrity, as institutions are rewarded for contributing to the governance process.

Despite its strengths, the system does have areas for improvement. During the test run, the issuance of duplicate certificates highlighted the need for enhanced validation mechanisms, including salting, to detect and prevent redundant entries. Future enhancements should also optimize gas fees, improve transaction times, and refine the user experience to ensure smooth, scalable operation.

In conclusion, "Immortal Certificates" is a robust, secure, decentralized digital credential management solution. Its strong emphasis on cybersecurity, combined with blockchain's immutability and IPFS's decentralized storage, positions it as a forward-thinking solution for ensuring the integrity and availability of digital certificates. As the project evolves, it will continue to enhance security and efficiency, offering a reliable academic and professional certification system.

11. References

[1] A. H. Monrat, O. Schel´en, and K. Andersson, "Docschain: Blockchain Based IoT Solution for Verification of Degree Documents", IEEE Trans. Comput. Soc. Syst., vol. 7, no. 3, Jun. 2020, pp. 561-572.

[2] G. Habib, S. Sharma, S. Ibrahim, I. Ahmad, S. Qureshi, and M. Ishfaq, "Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing", Future Internet, DOI: 10.3390/fi14110341.\, vol. 14, no. 11, Nov. 2022, pp. 341.

[3] Etherscan.io, "Ethereum Supply Statistics", https://etherscan.io/stat/supply. (July 2024).

[4] W. Haouari, A. S. Hafid, and M. Fokaefs, "Vulnerabilities of smart contracts and mitigation schemes: A Comprehensive Survey", Apr. 2024, preprint arXiv:2403.19805v2. https://arxiv.org/abs/2403.19805.

[5] Zhou, H., Milani Fard, A., Makanju, A. "The State of Ethereum Smart Contracts Security: Vulnerabilities, Countermeasures, and Tool Support", Journal of Cybersecurity and Privacy, vol. 2, no. 2, pp. 358-378, May 2022, DOI: 10.3390/jcp2020019.

[6] Cloudflare Developers, "Interplanetary File System (IPFS)", https://developers.cloudflare.com/web3/ipfs-gateway/concepts/ipfs/, 2023.

[7] P. Labs, "Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web", Aug. 2022, arXiv: 2208.05877,https://ar5iv.labs.arxiv.org/html/2208.05877.

[8] M. J. Rugaard, M. Egehave, and P. L. Damkjær, "Can NFTs be used to enforce trust in diplomas?", The Tokenizer, Sep. 19, 2022. Available: https://thetokenizer.io/NFT/can-nfts-be-used-to-enforce-trust-in-diplomas/

[9] S. T. Siddiqui, M. Fakhreldin, and S. Alam, "Blockchain Technology for IoT based Educational Framework and Credentials", in 2021 International Conference on Software Engineering Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), Pekan, Malaysia, 2021, pp. 194-199.

[10] Kabashi, F., Snopçe, H., Luma, A., and Neziri, V., "Trustworthy Verification of Academic Credentials through Blockchain Technology", International Journal of Online and Biomedical Engineering (iJOE), vol. 20, no. 9, 2024, pp. 51–64, DOI: 10.3991/ijoe.v20i09.48999.

[11] 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, vol. 14, no. 8, pp. 446, Aug. 2023, doi: 10.3390/info14080446.

[12] Ambast, S. K., Sumesh, T. A. "A Blockchain Based Credential Verification System using IPFS", IEEE 19th India Council International Conference (INDICON), DOI: 10.1109/INDICON56171.2022.10039743, 2022, pp. 1-6.

[13] T. Rahman, S. I. Mouno, A. M. Raatul, and N. Mansoor, "VerifiChain: A Credentials Verifier Using Blockchain and IPFS," in Information Systems and Applications, incl. Internet/Web, and HCI, Springer, Cham, 2023, pp. 246-261, doi: 10.1007/978-3-031-06188-8-19.

[14] Y. Wei, D. Trautwein, Y. Psaras, I. Castro, W. Scott, A. Raman, and G. Tyson, "The Eternal Tussle: Exploring the Role of Centralization in IPFS", NSDI 2024 Proceedings, Apr. 2024, pp. 1-15.

[15] W. Haouari, A. S. Hafid, and M. Fokaefs, "Vulnerabilities of smart contracts and mitigation schemes: A Comprehensive Survey", ResearchGate, Apr. 2024, https://www.researchgate.net/publication/379726984_Vulnerabilities_of_smart_contracts_and_mitigation_schemes_A_Comprehensive_Survey

[16] Lalli, A., Maciel, L., (2004) "A video demonstration of a test run for the DiplomaNFT Smart Contract

prototype". NYU Stream. https://agyacorp.com/Cybersecurity/0_-_Test_Run_Diploma_NFT_Team_A_wa_(HD_1080_-_WEB_(H264_4000)).mp4 (9 July 2004).
[17] Oladoja, I.P., Kolawole, O.H., Oronti, A. O., Abereowo, O. O., Akinsowon, O.A., Ogunlola, O. Y., and Alese, B. K., "Digital Academic Certificate Verification Using Blockchain Technology", World Congress on Education (WCE-2024), Churchill College, Cambridge University, UK, 26-28 August 2024.

12. Acknowledgements

We gratefully acknowledge the support and contributions of colleagues and collaborators in this research, especially the Tandon School of Engineering at New York University and the School of Computing and Data Science at Wentworth Institute of Technology.