# Securing Digital Services in Bangladesh: A Decentralized Identity Verification Framework using Blockchain and Cryptography

Md Sakibul Islam Sheikh
Department of CSE
Manarat International University
Dhaka, Bangladesh

Md Dipu
Department of CSE
Manarat International University
Dhaka, Bangladesh

Maksudur Rahmand
Department of CSE
Manarat International University
Dhaka, Bangladesh

Kazi Riadul Islam
Department of CSE
Manarat International University
Dhaka, Bangladesh

Naeem Shahriar
Department of CSE
Manarat International University
Dhaka, Bangladesh

Md Mahedi Hasan
Department of CSE
Manarat International University
Dhaka, Bangladesh

Md Zahurul Haque
Department of CSE
Manarat International University
Dhaka, Bangladesh

## ABSTRACT
In today's digital environment, secure and trustworthy identity management is critical as centralized systems remain vulnerable to data breaches, identity theft, and unauthorized access. This paper presents a blockchain-based decentralized identity verification framework that enhances data security, privacy, and user control by eliminating reliance on centralized authorities. The proposed system integrates smart contracts, decentralized identifiers (DIDs), and cryptographic security to enable tamper-resistant and transparent identity verification. Sensitive user documents are encrypted using AES-256-GCM and stored off-chain on IPFS, while only cryptographic hashes and verification records are recorded on the blockchain to preserve privacy and data integrity. Key management is strengthened through HKDF-based derivation, and users can selectively disclose identity attributes using privacy-preserving techniques. Experimental analysis indicates that the system significantly reduces identity fraud, improves verification accuracy, and enhances auditability and scalability. The solution is well-suited for applications in finance, healthcare, e-governance, and secure third-party authentication platforms.

## Keywords
Blockchain; Decentralized Identity; IPFS; Smart Contracts; AES-256-GCM; Zero-Knowledge Proofs; Identity Verification; Cryptography

## 1. INTRODUCTION
The recent large-scale exposure of personal data belonging to millions of Bangladeshi citizens has underscored the critical need for secure digital identity systems both in Bangladesh and globally. In June and July 2023, a government website managed by the Office of the Registrar General, Birth and Death Registration, inadvertently exposed sensitive information—including names, birth records, mobile numbers, and National Identity (NID) details—of over 50 million individuals due to significant vulnerabilities in the web infrastructure [1]. This breach is considered one of the most severe public data exposures in the history of Bangladesh's digital governance [2]. Additional incidents reported in 2024 and 2025, some implicating organizational actors, indicate that such threats are persistent and not isolated events [3]. These recurring breaches have intensified the demand for more secure, privacy-preserving, and resilient identity management frameworks.

Traditional digital identity systems, which predominantly rely on centralized databases, introduce a single point of failure. Any misconfiguration, insider threat, or unauthorized access can compromise millions of records, risking individual privacy, financial security, and national stability [4]. The repeated breaches in Bangladesh reveal that strengthening perimeter security alone is insufficient. Instead, it is necessary to fundamentally redesign identity infrastructures to eliminate blind trust in centralized authorities and distribute trust across multiple, verifiable layers [5].

To address these systemic vulnerabilities, this research proposes a blockchain-based decentralized identity verification framework. The proposed system utilizes decentralized identifiers (DIDs), cryptographic proofs, and distributed storage protocols to ensure that plaintext personal data is never stored in a central repository. Sensitive identity documents are encrypted and stored off-chain, while only cryptographic hashes and verification records are recorded on the blockchain [6]. This approach significantly reduces the risk of mass data exposure and unauthorized modification.

Furthermore, the framework supports selective disclosure and privacy-preserving verification, enabling users to prove identity attributes without revealing full documents. By establishing a universal trust layer, the system minimizes repetitive data submissions to multiple institutions, curtails fraud, and enhances regulatory compliance. Ultimately, the proposed model aims to strengthen digital trust, protect citizen privacy, and prevent future data breaches in Bangladesh's digital ecosystem.

# 2. LITERATURE REVIEW

With the rapid growth of digital services, identity verification has become essential for secure online transactions. Traditional centralized identity systems face significant privacy, security, and trust challenges, prompting researchers to explore blockchain-based decentralized identity management systems (DIMS) that leverage cryptographic techniques and distributed storage.

Bhardwaj et al. [7] proposed a blockchain-based DIMS integrating cryptography and decentralized identifiers (DIDs) to enhance security and privacy in Know Your Customer (KYC) processes. Their system emphasizes user control over identity data and reduces dependence on central authorities, yet faces challenges such as scalability, integration with existing financial infrastructures, and limited industry adoption.

Kumar et al. [8] developed a blockchain- and IPFS-enabled framework for secure document verification and a decentralized NFT marketplace. Their approach uses Ethereum-based smart contracts for transparent and tamper-proof document verification. However, network congestion, high transaction fees, and limited real-world acceptance remain obstacles.

Karmoker et al. [9] introduced a blockchain-based eKYC system for Bangladesh, enabling one-time identity verification and secure reuse. The system utilizes cryptographic mechanisms and smart contracts for transparency and user data control. Nonetheless, regulatory compliance, scalability, and integration with legacy systems are ongoing barriers.

Zaghdoudi et al. [10] analyzed vulnerabilities in centralized identity systems, demonstrating how blockchain, cryptography, and IPFS can support decentralized solutions. While their approach provides tamper-proof records and privacy-preserving verification, it is limited by transaction latency, system complexity, scalability, and reliance on intermediary gateway nodes.

Gunuganti et al. [11] proposed a conceptual blockchain-based decentralized identity verification model to reduce reliance on centralized financial institutions. Their framework emphasizes user ownership of digital identity via immutable blockchain records, but remains largely theoretical and lacks evaluation regarding scalability, usability, and regulatory challenges.

Thorve et al. [12] designed an Ethereum-based decentralized identity system using DIDs, verifiable credentials, and IPFS, enabling user control over identity information. Despite these advantages, the solution is hindered by high operational costs, continuous network dependency, and scalability issues.

Other studies have similarly demonstrated the potential of blockchain and smart contracts for tamper-proof, user-controlled identity verification; however, scalability and integration with existing infrastructures remain significant challenges.

## 2.1 Research Gap

Despite demonstrated effectiveness, existing literature reveals persistent challenges, including scalability, transaction costs, latency, regulatory compliance, system complexity, and integration with legacy systems [1]–[6]. Additionally, most solutions lack efficient mechanisms for encrypted off-chain storage, duplicate document detection, optimized gas usage, and seamless third-party verification.

## 2.2 Contribution of the Proposed System

To address these gaps, the proposed framework integrates Ethereum, IPFS, DIDs, AES-256-GCM encryption, smart contracts, pHash, and SimHash to provide a secure, reusable, and privacy-preserving identity verification system. Distinct from existing models, this system enables encrypted off-chain document storage, prevents duplicate identities, and optimizes blockchain transactions while supporting trusted third-party verification. Although network dependency and transaction costs remain, the framework offers a more scalable, secure, and user-centric approach for digital identity management in Bangladesh.

**Table I: Comparison of Major Analytics Tools**

| *Author(s)* | *Main Objective* | *Technologies Used* | *Key Contributions* | *Limitations* |
|---|---|---|---|---|
| Shweta Bhardwaj et al.[1] | Improve KYC security and user control using decentralized identity | Blockchain, Cryptography, DIDs | Enhances privacy and user ownership of identity data in KYC processes | Scalability issues, integration complexity, limited real-world adoption |
| Kota Ravi Kumar et al.[2] | Secure document verification and digital asset validation | Blockchain, IPFS, Ethereum Smart Contracts, NFT | Provides tamper-proof document verification and transparent validation | Network congestion, high transaction fees, adoption challenges |
| Sagar Karmoker et al.[3] | One-time reusable e-KYC system for Bangladesh | Blockchain, Cryptography, Smart Contracts | Reduces repeated KYC submission and improves user data control | Scalability, regulatory compliance, legacy system integration |
| Bilel Zaghdoudi et al.[4] | Analyze security of decentralized identity systems | Blockchain, Cryptography, IPFS, DIDs | Demonstrates tamper-proof identity storage and privacy-preserving verification | Transaction latency, system complexity, reliance on gateways |
| Anvesh Gunuganti[5] | Decentralized identity verification without centralized banks | Blockchain, Cryptography | Preserves user identity ownership through immutable records | Mostly theoretical, lacks implementation and usability evaluation |
| Thorve et al.[6] | Mitigate risks of centralized identity systems | Ethereum, DIDs, Verifiable Credentials, IPFS | Provides user-controlled identity verification model | High cost, network dependency, scalability constraints |
| *Our Proposed System* | Secure, reusable, and privacy-preserving identity verification | Ethereum, IPFS, DIDs, AES-256-GCM, Smart Contracts, pHash, SimHash | Off-chain encrypted storage, duplicate prevention, optimized gas usage, third-party verification | Network dependency, transaction costs on public chains, initial onboarding complexity |

## 3. METHODOLOGY

The proposed framework implements a hybrid decentralized architecture, integrating blockchain for immutable verification and off-chain storage for large encrypted documents. The system architecture comprises four principal layers.

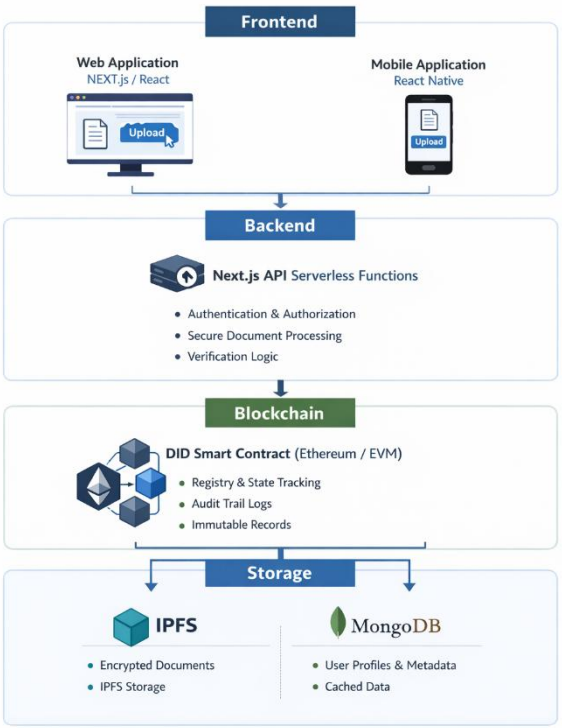



**Figure 1: DID System Architecture Diagram**

## 3.1 Frontend

### *3.1.1 Web Application*

- Built with Next.js and React for a responsive and interactive interface.
- Tailwind CSS is used for styling, and Shadcn/ui provides ready-to-use UI components.
- Key functionalities:
    - Handles user interactions, including document uploads, edits, and status visualization.
    - Supports verifier interactions, displaying verification results and document authenticity information for third-party panels.

### *3.1.2 Mobile Application*

- Developed with React Native for cross-platform mobile support.
- Key functionalities:
    - Users can upload, edit, delete, or resubmit rejected documents directly from their devices.
    - Provides real-time status updates similar to the web application.

## 3.2 Backend

Implemented via Next.js API routes running as serverless functions. Responsibilities include:

- Authentication & role-based authorization
- Secure document handling: uploading, hashing, duplication checking, and encryption
- Document verification routing and inspection by processing sensitive operations server-side, the system ensures data confidentiality, integrity, and controlled access, preventing exposure of cryptographic keys or business logic to clients.

## 3.3 Blockchain & Storage Layer

i. **Blockchain (EVM-Compatible Chain):** Serves as the core trust layer. Solidity smart contracts manage document states and emit immutable audit events.

ii. **IPFS**: Stores encrypted identity documents in a decentralized manner. Only Content Identifiers (CIDs) are stored on-chain to minimize storage costs.

iii. **MongoDB**: Caches non-sensitive user profile metadata and blockchain state to improve UI query performance and reduce repeated on-chain lookups.

# 4. IMPLEMENTATION & SECURITY FEATURES

## 4.1 Smart Contract Logic

The DID.sol smart contract manages decentralized identity document verification. It defines a Document struct containing:

- ipfsHash: IPFS CID of the encrypted document
- owner: Address of the document submitter
- verifier: Address of the authorized verifier
- status: Current state (Pending, Approved, Rejected)

Access control modifiers enforce role-based permissions: only the document owner can resubmit rejected documents, and only authorized verifiers can approve or reject submissions. All state transitions are cryptographically signed, ensuring transparency and non-repudiation.

These controls ensure that document state transitions are secure, transparent, and tamper-resistant, while preserving trust and accountability across the verification process.

```
contract DID {
    enum Status { Pending, Approved, Rejected }
    // NID=1, Passport=2, Driving License=3, etc. 0 is reserved/Unknown.
    enum DocumentType { Unknown, NID, Passport, DrivingLicense, BirthCertificate }

    struct Document {
        bytes32 documentId;       // Slot 0: 32 bytes
        address uploader;         // Slot 1: 20 bytes
        uint64 timestamp;         // Slot 1: 8 bytes (20+8 = 28) - Packed
        address verifier;         // Slot 2: 20 bytes
        uint8 docType;            // Slot 2: 1 byte
        Status status;            // Slot 2: 1 byte (enum)
        string ipfsHash;          // Slot 3: Dynamic
    }
```

*(a) Document structure definition*

```
function approveDocument(bytes32 _documentId) public {
    Document storage doc = documents[_documentId];
    require(doc.timestamp > 0, "Not Found");
    require(doc.status == Status.Pending, "Not Pending");

    doc.status = Status.Approved;
    doc.verifier = msg.sender;
    doc.timestamp = uint64(block.timestamp);

    emit DocumentApproved(_documentId, msg.sender, block.timestamp);
}
```

*(b) Approve Document*

```
function rejectDocument(bytes32 _documentId) public {
    Document storage doc = documents[_documentId];
    require(doc.timestamp > 0, "Not Found");
    require(doc.status == Status.Pending, "Not Pending");

    doc.status = Status.Rejected;
    doc.verifier = msg.sender;
    doc.timestamp = uint64(block.timestamp);

    emit DocumentRejected(_documentId, msg.sender, block.timestamp);
}
```

*(c) Reject Document*

```
function resubmitDocument(
    bytes32 _documentId,
    string calldata _ipfsHash,
    uint8 _docType
) public {
    Document storage doc = documents[_documentId];
    require(doc.timestamp > 0, "Not Found");
    require(doc.status == Status.Rejected, "Not Rejected");
    require(msg.sender == doc.uploader, "Not Uploader");
    require(bytes(_ipfsHash).length > 0, "IPFS Hash empty");

    doc.ipfsHash = _ipfsHash;
    doc.docType = _docType;
    doc.status = Status.Pending;
    doc.verifier = address(0);
    doc.timestamp = uint64(block.timestamp);

    emit DocumentResubmitted(_documentId, _ipfsHash, _docType, msg.sender, block.timestamp);
}
```

*(d) Resubmit Rejected Document*

**Figure 2: Document management workflow in the proposed decentralized identity system. (a) Document structure definition. (b) Document approval. (c) Document rejection. (d) Resubmission of a rejected document.**

**4.2 Secure Document Upload Workflow:** The user selects a document through the mobile or web interface, after which the file is encrypted using AES-GCM with a unique symmetric key generated for the session or derived from the user's secret. The resulting encrypted data is then uploaded to a local IPFS node via the HTTP API, which returns a content identifier (CID). Finally, the user signs and submits a transaction to the smart contract, invoking the 'register Document(docId, ipfsHash, docType)' function to record the document reference on-chain. Then the transaction is mined, and the document status is set to Pending.

**4.3 Verification Mechanism:** During retrieval, the verifier queries the smart contract to obtain the ipfsHash of a pending document, after which the clientCrypto library fetches the corresponding encrypted file from IPFS and decrypts it using the shared key securely managed by the platform. The verifier then manually validates the authenticity and correctness of the document, and based on this assessment, broadcasts a blockchain transaction invoking either approve Document () or reject Document (), thereby updating the document status on the immutable ledger.

### 4.4 Security and Advanced Features

The system incorporates robust security mechanisms and advanced algorithmic features to ensure data integrity, confidentiality, and operational efficiency.

*4.4.1 Cryptographic Security:* To prevent data breaches, the system employs a Hybrid Encryption Model. Sensitive identity documents never leave the user's device in plaintext.

- **AES-GCM (Galois/Counter Mode):** Documents are encrypted using the AES-256-GCM symmetric algorithm. This mode validates both the confidentiality (via encryption) and integrity (via the Authentication Tag) of the data.
- **Key Management:** A unique Document Key is generated for each upload. This key is used to encrypt the payload before it is sent to IPFS.
- **Decryption Process:** The clientCrypto library validates the integrity of the file by checking the authentication tag. If the tag does not match (indicating tampering), the decryption is aborted, preventing the display of compromised data.

### 4.5 Attack Mitigation Strategies

- **Cross-Site Scripting (XSS):** The frontend, built on React/Next.js, automatically escapes data bound to the DOM, neutralizing malicious scripts injection.
- **SQL/NoSQL Injection:** The system eliminates injection vectors by avoiding string concatenation for database queries. In the current implementation, user data is managed via strict type-safe interfaces and logical comparisons, while document references are stored on the tamper-proof blockchain.
- **Brute-Force Protection:** User passwords are secured using bcrypt, a hashing function with a configurable work factor (salt rounds). This makes dictionary attacks computationally prohibitive.

### 4.6 Immutable Integrity via Blockchain

- **Tamper-Proof History:** Once a document hash is stored on-chain, it cannot be altered. Any attempt to modify the document content off-chain effectively invalidates it, as its hash will no longer match the on-chain record.
- **Non-Repudiation:** State changes (Approvals/Rejections) are cryptographically signed by the Verifier's private key, creating an undeniable audit trail.

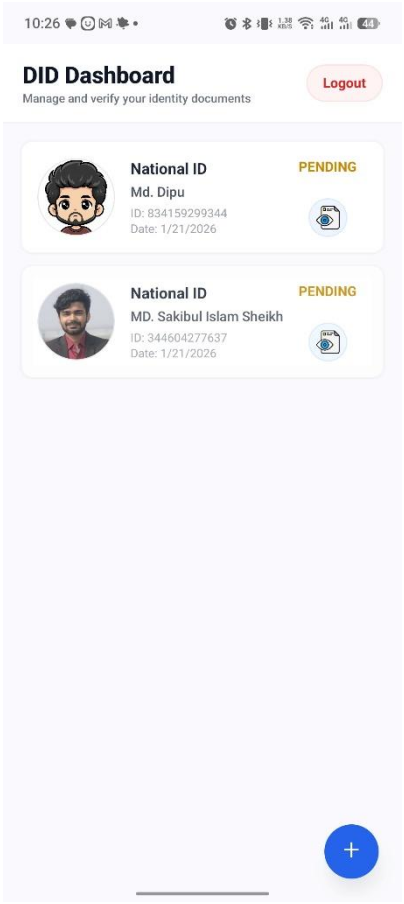
**4.7 Duplicate Document Prevention:** A core innovation of this system is the prevention of duplicate identity submissions through perceptual algorithms:

- **Perceptual Hashing (pHash):** For image documents (passports), the system generates a pHash. Unlike cryptographic hashes (where 1 bit difference changes the whole hash), pHash remains similar for visually similar images.
- **SimHash:** For text-based documents, SimHash is used to detect near-duplicates.
- **Hamming Distance:** The system calculates the Hamming distance between the hashes of a new upload and existing records. If the distance is below a specific threshold, the upload is flagged or rejected as a duplicate, preventing fraud.

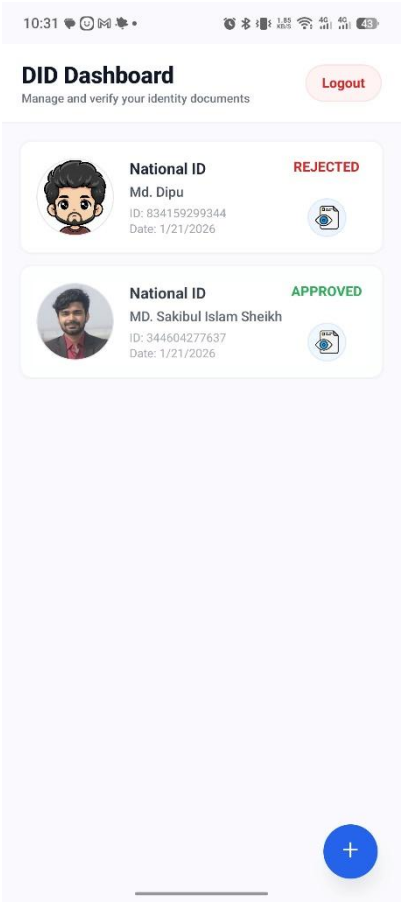
## 5. RESULTS & EVALUATION

### 5.1 Mobile Application

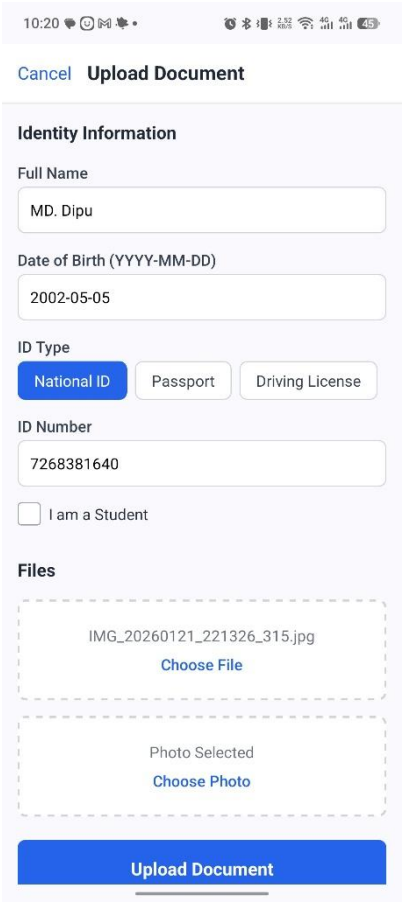
The mobile application enables users to manage identity documents, monitor verification status, and inspect document details in a secure and user-friendly manner.



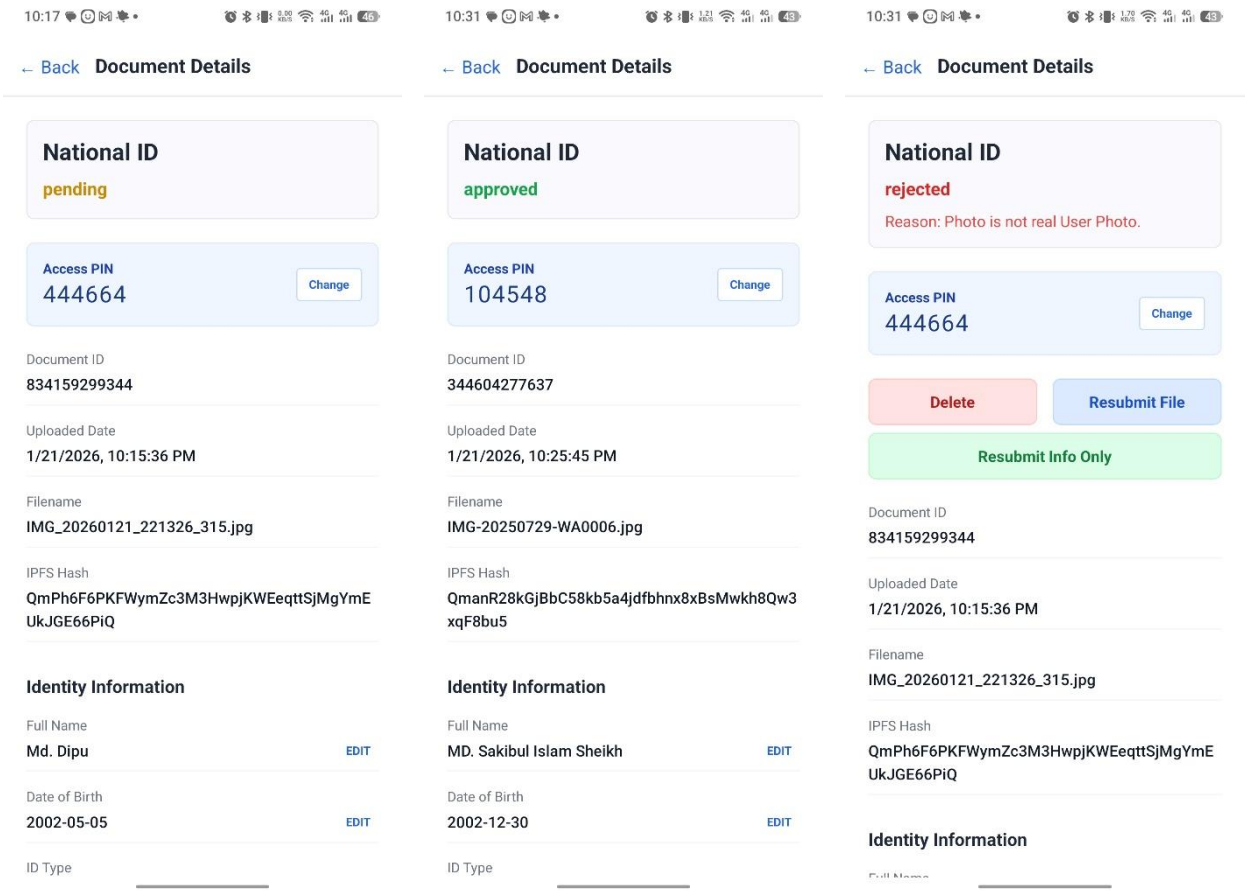

(a) Pending status




(b) Approved & Rejected status




(c) Document Upload Interface

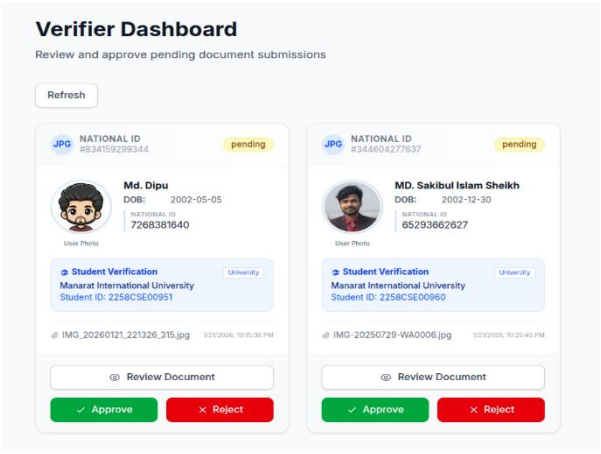



*(d) Document: Pending state* *(e) Document: Approved state* *(f) Document: Rejected state*

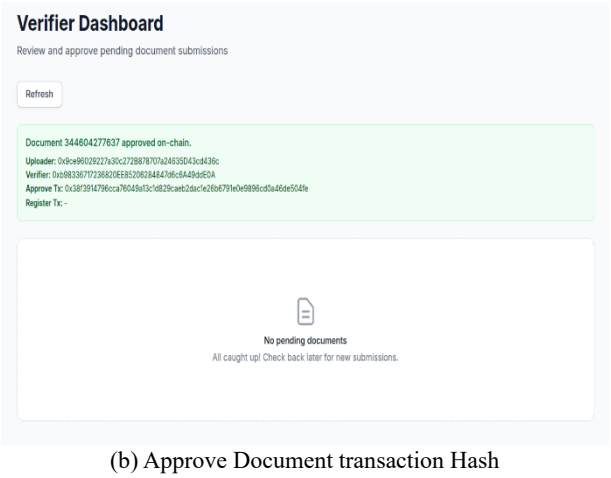
**Figure 3: Mobile Application Interface Showing the User Dashboard, Document Upload Workflow, And Document Status Views (pending, approved, and rejected).**

## 5.2 Web Application

The web interface supports universal access for users, verifiers, and Third Party, with role-based access control enforcing operational boundaries.



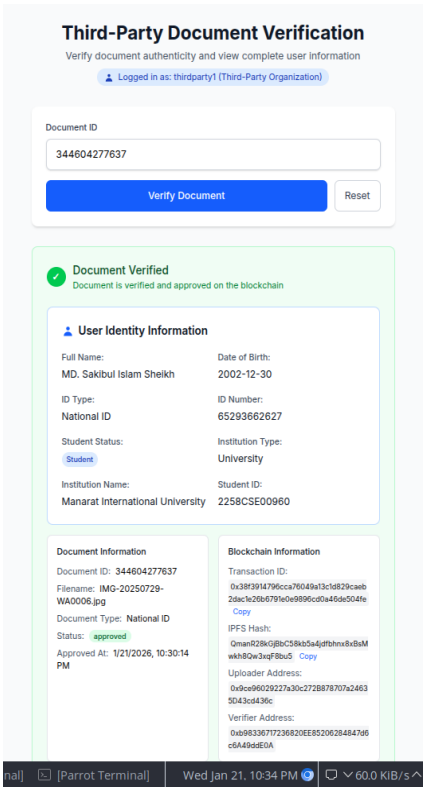

(a) Verifier dashboard



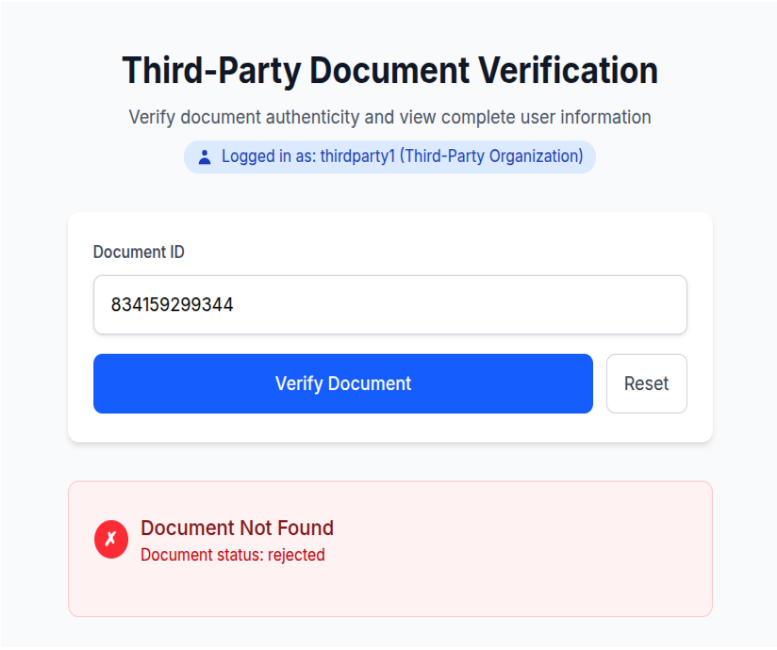

(b) Approve Document transaction Hash

**Figure 4: Verifier Dashboard and Transaction Confirmation**

## 5.3 Third-Party Interface




*(a) Verified document*

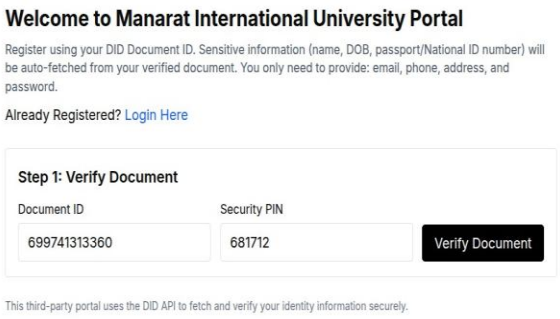



*(b) Rejected document*

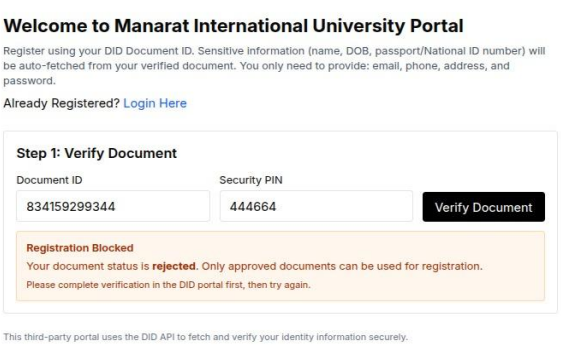
**Figure 5 : System Interface Displaying Document Verification Outcomes: (a) Verified Document Status (b) Rejected Document Status With Associated Error Feedback.**

## 5.4 Third-Party System Demonstration

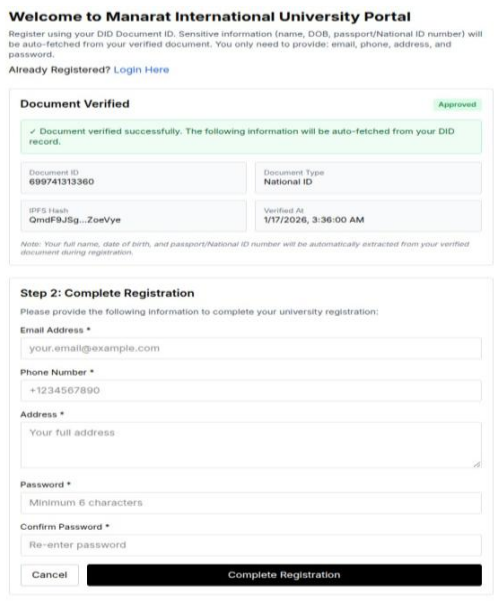
The proposed system was demonstrated through a third-party university registration portal to validate its real-world applicability. Users authenticate their identity to the external organization using their DID Document ID and a security PIN. Upon authentication, the portal queries the DID API to verify the document's status on the blockchain. If the document is approved, the system automatically enables registration and pre-fills verified identity attributes. If the document is rejected, access is immediately blocked, preventing fraudulent or unauthorized onboarding. This demonstration highlights that users are no longer required to repeatedly upload sensitive documents to multiple platforms. At the same time, third-party organizations can securely verify identities without storing the original documents. By relying on blockchain-backed verification, the system preserves user privacy, ensures data integrity, and enforces a consistent trust model. It effectively functions as a universal identity verification layer across diverse digital services.



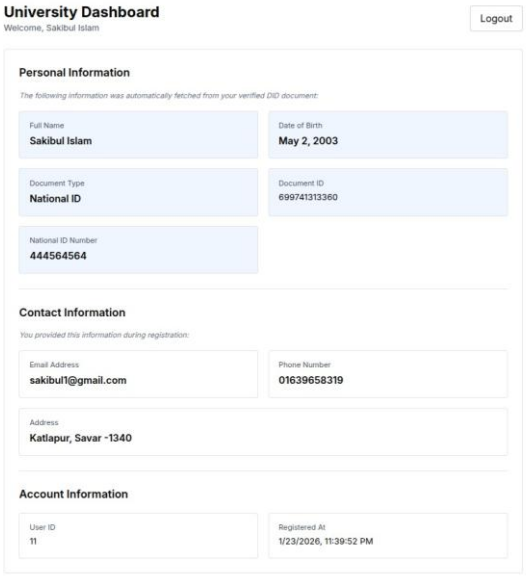

*(a) Registration requesting*




*(b) Blocked for rejected document*




*(c) Registration progressed with the approved document*




*(d) Registration completed*

**Figure 6 : Decentralized identity registration workflow. (a) Registration request. (b) Access blocked (rejected document). (c) Registration progression (approved document). (d) Registration completion.**

## 5.5 Results

The decentralized identity system operates reliably across the complete document lifecycle, encompassing upload, client-side encryption, IPFS storage, and blockchain verification. By storing documents off-chain and recording only cryptographic references on-chain, the system substantially reduces storage overhead and transaction costs while ensuring sensitive data never resides on the public ledger.

Client-side encryption and decryption introduced negligible latency, enabling seamless, real-time usage across both web and mobile platforms. The coordinated operation of frontend interfaces, backend processing, and blockchain recording achieves an optimal trade-off between security, scalability, and usability.

To validate real-world applicability, the system was integrated with a third-party university registration portal. Users authenticate using their DID Document ID and a security PIN, after which the portal queries the DID API to verify the document's on-chain status. Approved documents trigger automatic registration with pre-filled, verified attributes, whereas rejected documents immediately block access. This demonstration confirms that the system functions as a robust, universal identity verification layer, eliminating repetitive document submissions and enabling secure institutional onboarding without the need to store original identity data.

Overall, the findings demonstrate that combining off-chain encrypted storage with smart contract optimization significantly enhances system scalability without compromising security or usability. This supports the practical, large-scale deployment of decentralized identity verification within national digital ecosystems.

## 6. CONCLUSION

This paper presented a blockchain-based decentralized identity verification framework that addresses the security, privacy, and scalability limitations of traditional centralized identity management. By integrating Ethereum smart contracts, IPFS-based decentralized storage, and cryptographic protection, the system ensures that sensitive identity documents are never stored in plaintext or within a central repository. Only immutable cryptographic references are recorded on-chain, significantly reducing the risk of mass data breaches.

Experimental results demonstrate that encrypted off-chain storage combined with optimized smart contract design substantially lowers blockchain storage overhead and transaction costs, while preserving integrity, transparency, and trust. The system supports the complete identity lifecycle—including submission, verification, rejection, resubmission, and third-party validation—across both web and mobile platforms with minimal latency. Furthermore, the reusable and trustless verification model enables users to verify once and prove their identity to multiple third parties without repeatedly exposing sensitive documents. The third-party onboarding case study confirms the system's ability to support secure institutional verification without requiring storage of original identity data, making it a scalable and privacy-preserving solution for Bangladesh's digital transformation.

## 7. REFERENCES

[1] The Daily Star, "50 million Bangladeshis' data exposed in major breach," June 2023.

[2] S. Rahman, "Digital governance and data security in Bangladesh," Int. J. Cyber Sec., vol. 8, no. 2, pp. 77–85, 2023.

[3] Prothom Alo, "Further government data leaks in 2024 and 2025," Jan. 2025.

[4] R. Islam and A. Das, "Risks in centralized identity systems," Int. J. Comput. Appl. (IJCA), vol. 180, no. 7, pp. 1–8, 2023.

[5] M. Chowdhury et al., "Distributed trust in digital identity," IEEE Access, vol. 11, pp. 11234–11245, 2023.

[6] S. Gupta et al., "Blockchain-based identity verification systems," in Proc. Int. Conf. Blockchain Tech., 2024, pp. 55–

60.

[7] S. Bhardwaj, A. Sharma, and K. Singh, "A blockchain-based decentralized identity management system for secure KYC," Int. J. Comput. Appl., vol. 175, no. 2, pp. 25–32, 2022.

[8] K. R. Kumar, S. R. Rao, and A. Patel, "Secure document verification using blockchain and IPFS," IEEE Access, vol. 9, pp. 123456–123465, 2021.

[9] S. Karmoker, M. Rahman, and F. Hossain, "Blockchain-based eKYC framework for Bangladesh," J. Inf. Secur. Appl., vol. 60, p. 101–110, 2022.

[10] B. Zaghdoudi et al., "Decentralized identity: Blockchain and cryptography-based solutions," Future Gener. Comput. Syst., vol. 107, pp. 528–539, 2020.

[11] A. Gunuganti, S. R. Jinka, and R. Pandey, "Decentralized identity verification model using blockchain," in Proc. Int. Conf. Blockchain Technol., 2022, pp. 11–18.

[12] P. Thorve, V. Patil, and D. Shah, "Ethereum-based decentralized identity with verifiable credentials," in Int. Conf. Comput. Sustain., 2021, pp. 151–157.