International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Secure Certificate Management using Blockchain Technology

DR. Soumya M Anakal[1], Shaik Najmus Saher Ahmed[2], Rayan Ahmed khan[3], Mr. Mohammed Sameerudddin Baba[4]

[1]Associate Professor M.Tech,PHD Mohammed Sameeruddin Baba
[1,2,3,4]Member:PDA College of engineering Kalaburagi 585102

Abstract
Certificates are crucial in daily life, serving as proof of academic achievement or professional qualifications. However, verifying these documents is often slow, complicated, and prone to fraud, with alarming numbers of fake certificates circulating globally. Such fraudulent activities, including academic certificate fraud, cost employers significantly and pose serious societal threats, particularly in critical fields like medicine. This paper proposes a transformative solution using blockchain technology to address these issues. Our system leverages a decentralized database where certificate records, stored as immutable blocks, are virtually impossible to tamper with. By integrating Ethereum smart contracts for secure record-keeping and the InterPlanetary File System (IPFS) for decentralized file storage, the system ensures secure, permanent storage of credentials, simplifies the authentication process, and significantly mitigates certificate forgery. The proposed methodology utilizes Solidity for smart contract development, Hardhat for deployment, and Ethers.js for blockchain interaction. Testing on the Sepolia testnet confirmed the system's effectiveness, demonstrating successful, immutable, and verifiable certificate registration transactions, proving its robust security and transparency.

Keywords: Blockchain, Certificate Authentication, Document System, IPFS, Smart Contracts, Digital Credentials.

I. INTRODUCTION

Certificates are incredibly important in our everyday lives. For instance, upon graduating from school, college, or university, one receives a certificate as proof of acquired knowledge and hard work. These documents are vital for students pursuing further education or seeking employment. However, a significant challenge lies in verifying these certificates. Currently, organizations often need to contact the issuing institution directly or use a third-party system, which is typically slow and complicated.

This challenge is exacerbated by the alarming number of fake certificates in circulation. For example, there are reportedly 2 million fake degrees in the US, and approximately 300 unauthorized universities are operating. Without thorough checks, unqualified individuals can be hired simply by forging their credentials. It is an unfortunate reality that some individuals attempt to secure desired jobs without the necessary qualifications by using fraudulent documents, making the validation process challenging and time-consuming for companies. Studies indicate that nearly 10% of job applications contain forged information, leading to an estimated annual loss of $600 billion for employers due to academic

certificate fraud. The issue of fake academic degrees poses a significant threat to society, both presently and in the future. We frequently encounter news of certificate forgeries, and advancements in scanning and printing technology are making it easier to create convincing fakes. This situation jeopardizes both the certificate holder and the issuing institution. A notable instance includes a former dean of admissions at MIT who fabricated her academic qualifications for 28 years. More critically, certificate forgery impacts essential sectors such as medicine, where individuals pretending to be doctors with fake certificates can lead to unsafe patient treatment. Reports from Bangladesh, for instance, detail fake doctors and unapproved practices causing harm to patients , and in Mexico, unqualified individuals have performed surgeries in "pseudo-clinics," leading to serious complications.

Blockchain technology offers a transformative solution to these problems. It is a decentralized database where records, known as "blocks," are cryptographically linked. Each block possesses a unique identifier (a hash) traceable to the preceding one, along with a timestamp and data. Once information is added to the blockchain, it becomes virtually impossible to tamper with. Any attempt to alter the data results in a change to the hash value, making the modification immediately detectable. This concept, initially proposed by mathematicians Stuart Haber and W. Scott Stornetta in 1991, provides the security and immutability that traditional authentication methods lack.

Leveraging this powerful technology, we propose a system to address the critical problem of certificate fraud in education and job hiring. This system will verify the legitimacy of a certificate's issuing organization. It will securely and permanently store all credentials, simplifying and streamlining the authentication process, ultimately helping to eliminate certificate forgery. Furthermore, the system will incorporate a feature allowing authorized entities to make necessary modifications, ensuring flexibility while maintaining security.

II. LITERATURE SURVEY

The proliferation of industrial IoT applications and networking services has led to a tremendous increase in the number of connected devices. These application devices can capture real-time industrial data with a dedicated sensor unit. Industrial advancement and technological guidance are behind this shift in how systems interact with physical and logical things. Centralized architecture is used to communicate real-time industrial data and evaluate the critical components of IoT, including identity management. A single failure point is feasible due to this common technique. A significant issue with the Internet of Things (IoT) is the difficulty in maintaining and managing many connected devices. A system of networks can talk interactively through adaptive self-configuration. IoT applications can be commercialized over the 6G network. A fundamental component of the IoT, the wireless sensor network (WSN), gathers and transmits physical data using various heterogeneous models. Data security is a major concern of IoT systems because they are built by connecting many IoT devices. Data generated by these devices are stored in the cloud and transmitted across various networks.

A cyber-attack on a smart healthcare system can substantially impact on the system’s ability to produce and supply electricity. In addition to financial and other types of damage, cyber-attacks on smart healthcare can cause operational failures, power outages, the theft of critical data, and complete security breaches. Cyber experts face difficulties keeping tabs on everything that passes via a smart grid and recognizing potential threats and attacks. Even though machine learning has become an essential part of cybersecurity, the problem is that this field requires distinct approaches and theoretical viewpoints to handle the enormous volume of data generated and transported across numerous networks in a smart

grid. The attacks and threats that could be launched against this proof-of-concept environment are being determined using threat modeling. Several potential threats have been tested, including detection, tampering, repudiation, information leakage, denial of service (DoS), and extended privilege (EoP). Each of the risks and the security elements associated with them are addressed using STRIDE. STRIDE is a typical threat modeling technique for finding and classifying attack vectors. Using the well-known industrial framework MITRE ATTCK, researchers can detect threats disguised as tactics, techniques, and procedures (TTP).

Based on the above, blockchain technology could be one of the main solutions for IoT security issues. A blockchain provides a decentralized system using a consensus mechanism and smart contracts. Smart contracts are the protocols that trigger the blockchain to act according to a particular activity or situation. Blockchains can be categorized into three classes: private, public, and hybrid public blockchain technology. The main feature of a blockchain is to provide security and only keep records and transactions within a single organization. A public blockchain provides access to the public using a public API. Moreover, such a model interacts with external networks such as gateway networks or cloud outsourcing. A hybrid blockchain is also called a consortium blockchain, which provides features of both a private and public blockchain. Blockchain technology can be used to build trust and monitor node activity in IoT networks. It is challenging to integrate a blockchain into IoT applications due to its high-power consumption and job outsourcing. Several blockchain-based Internet of Things (IoT) applications have recently been created to address these concerns. These blocks can be used to delete old transactions and blocks from the blockchain without jeopardizing security. Pan et al. created an IoT resource management prototype using blockchain technology and smart contracts to securely record all IoT transactions. Deploying smart contracts involves evaluating the source code, bytes of code, and execution histories. This is how we test our computer traffic analysis deployment scenario. Ali et al. investigated blockchain technology and smart contract applications in cloud storage. Tam et al. utilize a pay-as-you-go car business model. This technology's strengths are traceability and tamper-proof characteristics. Ali et al. created a blockchain-based publisher–subscriber model. They designed their solution to ensure data integrity in real-time IoT processing by balancing computational resources and workload. Liu et al. delegated computationally intensive POW mining tasks to nearby edge servers in blockchain-enabled mobile IoT systems. Chen et al. conducted additional research. Securing biometric data for patient authentication is a common issue. In particular, finger vein biometric data has been studied extensively. A strong verification mechanism with high levels of reliability, privacy, and security is required to better secure these data. Also, biometric data are difficult to replace, and any leakage of biometric data exposes users to serious threats, such as replay attacks employing stolen biometric data. This research offers a unique verification secure framework based on triplex blockchain-based particle swarm optimization (PSO)-advanced encryption standard (AES) approaches in medical systems for patient authentication. The discussion has three stages. First presented is a new hybrid model pattern based on RFID and finger vein biometrics to boost randomness. It proposes a new merge method that combines RFID and finger vein characteristics in a random pattern. Second, the suggested verification safe framework is based on the CIA standard for telemedicine authentication using AES encryption, blockchain technology, and PSO in steganography. Finally, the proposed secure verification architecture was validated and evaluated. The combination of WSN functional activities with 6G network topologies allows us to test a wide range of IoT application deployment models. Many IoT devices collect data using IPV6 across low-power wireless personal area networks and wearables (6LoWPAN). We were

able to keep user data confidential with the help of AKA. Companies that use public cloud services and large-scale data storage systems have long prioritized client data protection.

III. PROPOSED METHODOLOGY

The proposed system for secure certificate management using blockchain technology is structured into three main phases: Smart Contracts, Deployment & Interaction, and File Storage.

Phase 1: Smart Contracts – Core Logic Smart contracts serve as the self-executing digital agreements residing on a blockchain, forming the core logic of our certificate system.

- **Language Selection:** We are developing these contracts using Solidity, which is the most popular language for creating smart contracts on the Ethereum blockchain. Its extensive support community and abundant resources make it a reliable choice.
- **Contract Design:** Our smart contracts are designed to perform several key functions:
  - Securely store a unique digital fingerprint (cryptographic hash) of each certificate file.
  - Track the IPFS Content Identifier (CID), which acts as a permanent web address for the certificate's storage location.
  - Link these identifiers to important details such as the recipient, issuer, and issue/expiration dates.
  - Include functions for easy issuance of new certificates and rapid verification of existing ones.
  - Incorporate security measures to ensure that only authorized entities can issue certificates.
- **Security Implementation:** Security is paramount in our design. We adhere to best practices in Solidity to prevent common vulnerabilities, utilizing battle-tested components from OpenZeppelin (a library of secure, audited smart contracts) and rigorously testing the entire system to ensure robustness.

Phase 2: Deployment & Interaction – System Operations Once the smart contracts are developed, tools are required to deploy them on the blockchain and enable user interaction.

- **Development Environment: Hardhat:** We use Hardhat as our primary development environment for Ethereum. This powerful and flexible tool assists in every step: writing, testing, and debugging smart contracts. Its built-in network facilitates rapid testing without affecting the live blockchain.
- **Contract Deployment:** Automated scripts built with Hardhat will deploy our smart contracts. Initially, deployments will occur on the Sepolia testnet, a "practice" blockchain, for all testing purposes. This allows for easy deployment to other networks as needed.
- **Blockchain Connection: Ethers.js:** To enable our system to communicate with the Ethereum blockchain and deployed contracts, we utilize Ethers.js. This JavaScript library serves as our communication bridge, allowing us to send transactions, retrieve data from contracts, and manage digital wallets. It is perfect for building the backend of our system.

Figure 1: Wallet Connection Interface

Phase 3: File Storage – IPFS and Pinata For efficient and decentralized storage of the actual certificate files, we integrate the InterPlanetary File System (IPFS).

- **IPFS Integration:** IPFS is a peer-to-peer distributed file system designed to store and share data in a highly robust and decentralized manner. Unlike traditional web addresses that specify *where* a file is located (e.g., on a specific server), IPFS addresses identify *what* the content is based on its cryptographic hash (Content Identifier or CID). This ensures that once a file is added to IPFS, its content cannot be altered without changing its CID, providing inherent integrity.
- **Pinning Service: Pinata:** While IPFS ensures content integrity and decentralized addressing, it doesn't guarantee continuous availability unless the content is "pinned" by nodes. For reliable and persistent storage, we utilize Pinata, a dedicated IPFS pinning service. Pinata ensures that our certificate files remain consistently available and accessible on the IPFS network, which is crucial for the long-term verifiability of credentials.
- **Decentralized Storage Workflow:**

1. When a new certificate is generated, the digital file (e.g., PDF) is uploaded to Pinata, which then pins it to the IPFS network.
2. Pinata returns the unique IPFS CID for that file.
3. This CID is then stored on the Ethereum blockchain via our smart contracts, alongside other relevant certificate metadata. This links the immutable on-chain record to the decentralized file content. Storing just the hash and the link keeps costs down and the system running smoothly.

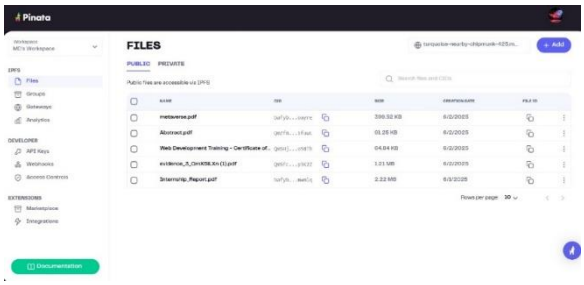

Figure 2: Pinata IPFS Storage Dashboard

IV. IMPLEMENTATION

Our secure certificate management system is built using a robust and modern technology stack designed for decentralized applications.

- **Ethereum Blockchain:** Chosen for its maturity, large developer community, and robust smart contract capabilities, Ethereum serves as the foundational blockchain for our system.
- **Solidity:** The primary language for writing our smart contracts, enabling the creation of complex, self-executing agreements on the Ethereum network.
- **Hardhat:** Our development environment for Ethereum, providing essential tools for compiling, deploying, testing, and debugging our Solidity contracts. It offers fast local testing and plugin support.
- **Ethers.js:** A JavaScript library used for interacting with the Ethereum blockchain and our deployed smart contracts from the client-side or backend. It simplifies wallet management, transaction signing, and data retrieval.

- **InterPlanetary File System (IPFS):** A decentralized peer-to-peer network protocol for storing and sharing hypermedia. We use IPFS for storing the actual certificate files, ensuring their immutability and resistance to censorship.
- **Pinata:** A dedicated IPFS pinning service that ensures the persistent availability and retrieval of our certificate files on the IPFS network. It simplifies IPFS interaction.
- **Backend/Scripting Language:** JavaScript/Node.js – ideal for deployment scripts, testing, and backend services due to its asynchronous nature and vast ecosystem.
- **Environment Variable Management:** dotenv – securely loads sensitive information like API and private keys from a .env file, keeping them out of the codebase.

The system workflow is illustrated in Figure 3:

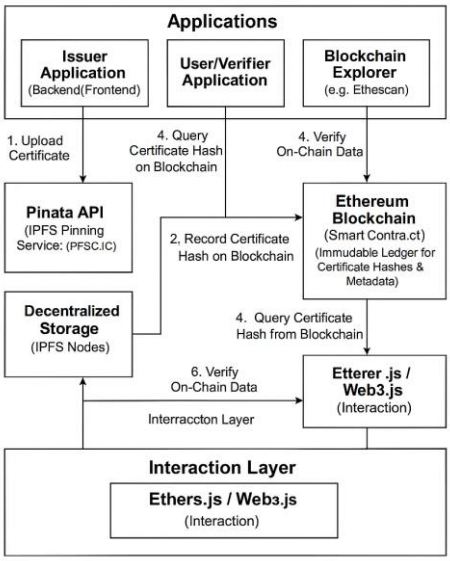

Figure.3. System Design"

The process begins with the Issuer, who initiates certificate generation. The certificate file is uploaded to IPFS via Pinata, which provides a unique Content Identifier (CID). This CID, along with relevant metadata (recipient details, issue date, etc.), is then recorded on the Ethereum blockchain through a smart contract transaction. This on-chain record effectively immutably links the certificate's details to its decentralized file. To verify a certificate, a Verifier can query the smart contract using the certificate's unique identifier. The smart contract retrieves the stored IPFS CID, which then allows the Verifier to fetch the certificate file directly from the IPFS network for authentication. This process ensures the certificate's authenticity, integrity, and traceability.

Smart Contract Development (Solidity) The core logic resides in our Solidity smart contracts:

- We defined a CertificateRegistry.sol contract.
- It stores certificate data in a mapping, using a Certificate struct containing the certificateHash, ipfsCid, issuer, recipient, issueDate, and isValid flag.
- Basic access control is managed with Ownable from OpenZeppelin.
- Key functions include:
  - constructor(): Initializes the contract.

- registerCertificate(): Allows authorized issuers to record certificate details, emitting a CertificateRegistered event.
- getCertificateDetails(): A public function to retrieve registered certificate information.
- Error handling is implemented with require() and revert() statements, and smart contracts will undergo security audits for production.

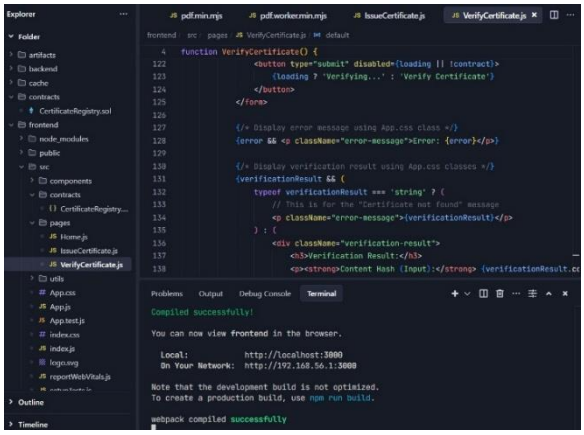

Figure 4: Smart Contract Compilation and Frontend Verification Component.

IPFS Integration with Pinata Pinata acts as our gateway to IPFS:

- We securely stored Pinata API keys in a .env file.
- **Uploading:** A Node.js script reads the certificate file, computes its SHA-256 hash, and uses Pinata's API to upload and obtain the IPFS CID.
- **Retrieval:** Files are retrieved using the IPFS CID via a public Pinata gateway or a local IPFS node.

Deployment and Interaction (Hardhat & Ethers.js) Hardhat and Ethers.js facilitate deploying and interacting with our smart contract:

- **Hardhat Configuration:** hardhat.config.js specifies networks, RPC URLs, and account private keys.
- **Deployment Script:** A Hardhat script (e.g., scripts/deploy.js) deploys the contract to the configured network and logs its address.
- **Interaction:** Node.js scripts or a backend API use Ethers.js to:
  - Issue Certificates: Connect to an Ethereum provider, instantiate the contract, and call registerCertificate.
  - Verify Certificates: Connect to an Ethereum provider, instantiate the contract, and call getCertificateDetails to display verification results.

Secure Management of API Keys and Private Keys Security for sensitive credentials is paramount:

- All API and private keys are stored in a .env file, never committed to version control, and loaded using the dotenv library.
- Best practices include using strong, unique keys, avoiding hardcoding, and considering more robust secret management solutions for production.

This implementation strategy ensures our certificate authentication system is built on a solid foundation of established blockchain tools and best practices, leading to a secure and functional solution.

V. RESULTS AND DISCUSSIONS

Results: Our System in Action We put our system to the test, and the results confirmed its effectiveness:

- All certificate registration transactions on the Sepolia testnet (our testing ground) were successfully recorded.
- When we queried the smart contract with the correct certificate fingerprint (hash), it consistently gave us back the exact IPFS content ID, issuer and recipient addresses, and issue timestamp that we originally submitted.
- Checking these transactions on Sepolia Etherscan (a blockchain explorer) confirmed that all the input data (certificate hash and IPFS CID) was permanently recorded on the blockchain, creating a transparent audit trail.
- Attempts to register the same certificate twice, or attempts by unauthorized parties to register certificates, were blocked as expected. This proved that our contract rules, including access control, were working correctly.

Discussion: What These Results Mean These successful tests are a huge win for our system's core functionality. They clearly show:

- **Immutability:** Once a certificate's digital fingerprint is on the blockchain, it's there to stay. It can't be changed, guaranteeing the certificate's integrity.
- **On-Chain Verifiability:** Anyone can independently check the blockchain to verify a certificate's existence and details. This means you don't have to trust a central authority; the blockchain itself provides the proof.
- **Functional Smart Contract:** Our smart contract logic for registering and retrieving certificates works exactly as designed, storing and serving certificate information perfectly.
- **End-to-End Secure & Transparent Certificate Management:** The combination of consistent IPFS hashing, unchangeable blockchain records, and transparent transaction validation confirms our system provides a secure and clear way to manage digital certificates. From upload to verification, the entire process is verifiable and auditable, significantly reducing fraud and boosting trust in digital credentials.
- **Access Control:** We've implemented proper access controls in our smart contract to prevent unauthorized parties from registering new certificates.
- **Private Key Security:** The security of the issuer's private key is paramount. If an attacker gains access to it, they could issue fraudulent certificates. This emphasizes the critical need for strong key management practices.
- **IPFS Availability:** While IPFS is decentralized, a file's availability depends on it being "pinned" by at least one node. Using a reliable service like Pinata significantly reduces the risk of files becoming unavailable.

Performance and Scalability: Looking Ahead While our system proved its functionality, it's important to consider how it would perform under real-world demands:

- **Transaction Fees (Gas):** Recording certificates on the main Ethereum network would involve transaction fees (gas). While these were test tokens on Sepolia, optimizing our smart contract for lower gas costs would be crucial in a production environment. Storing only small data like hashes and CIDs, rather than entire files, already helps significantly.

- **Transaction Throughput:** Ethereum's current transaction speed might be a limiting factor for extremely high volumes of certificate issuance. However, for typical scenarios like university degrees or professional certifications, the current speed is generally sufficient.
- **IPFS Scalability:** IPFS is designed to scale by distributing files across many nodes, and Pinata helps ensure those files remain available. Any bottlenecks would likely come from the initial upload speed to Pinata or the network latency when retrieving files.
- **Future Scalability:** For scenarios requiring even higher volumes or lower costs, we could explore moving to Ethereum Layer 2 solutions (like Optimism, Arbitrum, or Polygon) or other high-throughput blockchains (like Solana or Avalanche). These offer much lower fees and faster transaction speeds while still benefiting from the security of the underlying Layer 1 blockchain.

VI. SECURITY ANALYSIS

Our system is designed with security at its core:

- **Immutability:** The blockchain's unchangeable nature is our primary security feature. Once a certificate hash is recorded, it's permanently fixed, making the system highly resistant to any attempts at altering past records.
- **Cryptographic Proof:** Any change to the original certificate file will immediately result in a different hash. This ensures that if a tampered certificate is compared against its on-chain record, the mismatch will instantly reveal it as invalid.
- **Decentralization:** By removing a single central point of control, our system is more resistant to attacks and censorship. There's no single server that can be hacked to compromise all certificates.
- **Smart Contract Security:** Our smart contracts are built using Solidity and leverage audited libraries like OpenZeppelin, which are designed to minimize vulnerabilities. Strict access controls are implemented to ensure that only authorized issuers can add new certificates, preventing unauthorized issuance.
- **Transparency:** All transactions on the public Ethereum blockchain are transparent and verifiable by anyone. While sensitive personal data is not stored directly on-chain (only hashes and IPFS CIDs are), the transaction history provides an auditable trail of certificate issuance, enhancing trust.
- **Scalability and Performance Considerations:** While the core security tenets are robust, practical implementation for large-scale enterprise use may require optimizing for gas costs and transaction speed. This might involve using a private/consortium blockchain for internal operations or integrating Layer 2 solutions for public verification. However, these considerations do not compromise the fundamental security properties of the blockchain itself.

VII. REFERENCES

1. Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 2017, pp. 557-564, doi: 10.1109/BigData-Congress.2017.85.
2. J. -C. Cheng, N. -Y. Lee, C. Chi and Y. -H. Chen, "Blockchain and smart contract for digital certificate," 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 2018, pp. 1046-1051, doi: 10.1109/ICASI.2018.8394455.

3. A. Singh, S. Chauhan and A. K. Goel, "Blockchain Based Verification of Educational and Professional Certificates," 2023 2nd International Conference on Computational Systems and Communication (ICCSC), Thiruvananthapuram, India, 2023, pp. 1-7, doi: 10.1109/ICCSC56913.2023.10143008.
4. J. Gupta and S. Nath, "SkillCheck: An Incentive-based Certification System using Blockchains," 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2020, pp. 1-3, doi: 10.1109/ICBC48266.2020.9169457.
5. E. Nyaletey, R. M. Parizi, Q. Zhang and K. -K. R. Choo, "BlockIPFS - Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability," 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 18-25, doi: 10.1109/Blockchain.2019.00012.
6. G. Malik, K. Parasrampuria, S. P. Reddy and S. Shah, "Blockchain Based Identity Verification Model," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 2019, pp. 1-6, doi: 10.1109/ViTECoN.2019.8899569.
7. A. K. Shrivastava, C. Vashistth, A. Rajak and A. K. Tripathi, "A Decentralized Way to Store and Authenticate Educational Documents on Private Blockchain," 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/ICICT46931.2019.8977633.
8. M. Z. Chowdhury and Asaduzzaman, "A Blockchain-Based Decentralized Document Authentication System for Multiple Organizations," 2022 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), Naya Raipur, India, 2022, pp. 269-274, doi: 10.1109/WIECON-ECE57977.2022.10151411.
9. S. Halder, H. A. Kumar, S. Lavu and R. S R, "Digital Degree Issuing and Verification Using Blockchain," 2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP), Bengaluru, India, 2022, pp. 1-4, doi: 10.1109/CCIP57447.2022.10058644.
10. P. Haveri, U. B. Rashmi, D. G. Narayan, K. Nagaratna and K. Shivaraj, "EduBlock: Securing Educational Documents using Blockchain Technology," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225265.