International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

Digital Locker using Block-Chain

Mrs. V. Roopa[1], Ms. Kaviyasri M [2], Ms. Pavithra D[3]
[1,2,3]Department of Cyber Security, Mahendra Engineering College, Namakkal, India.
Emails: *roopa1995@gmail.com[1], kaviyasrisri49@gmail.com[2], pavithradharmalingam3@gmail.com [3]*

Abstract
The Rapid digital transformation has created an urgent need for secure and reliable document management systems. Traditional digital storage methods are often vulnerable to tampering, unauthorized access, and single points of failure. This paper presents a Blockchain-based Digital Locker system designed to offer secure, decentralized, and tamper-evident storage for digital documents. Leveraging the inherent properties of blockchain immutability, transparency, and distributed consensus our system ensures the authenticity, integrity, and traceability of stored documents. The platform includes features such as role-based access control, cryptographic verification, and real-time audit trails. This Digital Locker not only empowers individuals with control over their personal documents but also facilitates secure document sharing and verification for institutions and service providers. The proposed system is a scalable, robust solution for digital document management in sectors such as education, healthcare, and governance.

Keywords: *Blockchain, Digital Locker, Document Verification, Decentralized Storage, Access Control, Data Integrity, Cryptography, Smart Contracts, Audit Trail*

1. Introduction

In the modern digital age, data security and trust in digital documentation have become central concerns for individuals, enterprises, and governments alike. With the proliferation of online services, an increasing volume of personal and institutional data is being shared, stored, and accessed digitally [1]. This includes sensitive documents such as identity proofs, academic certificates, health records, financial statements, contracts, and property documents. The transition from paper-based to digital documentation offers numerous advantages improved accessibility, convenience, and cost efficiency but also raises significant issues regarding data security, authenticity, and control. Traditional cloud-based or centralized document management systems, while widely used, are often limited in their ability to ensure data integrity, prevent unauthorized modifications, and provide a transparent audit trail. These systems typically rely on a single authority or entity to manage and validate documents. This centralization creates multiple challenges, such as vulnerability to cyberattacks, data manipulation by insiders, single points of failure, and a lack of user autonomy over their data. Additionally, verifying the authenticity of a document often involves time-consuming manual processes or reliance on trusted intermediaries. In response to these limitations, blockchain technology has emerged as a revolutionary paradigm offering decentralized, transparent, and secure solutions to data management challenges. Originally developed as the underlying technology for cryptocurrencies, blockchain has found increasing applications in various sectors including finance, supply chain, healthcare, and digital identity management. Its key features immutability, decentralization, cryptographic security, and consensus-driven validation make it an ideal candidate for building trustworthy digital ecosystems. This paper introduces a novel application of blockchain: a Digital Locker system that enables secure, tamper-proof, and decentralized storage and management of digital documents. Unlike traditional systems, the proposed solution empowers users by giving them full control over their data while eliminating the need to rely on centralized service providers for storage, validation, or sharing. The system leverages smart contracts to enforce document access permissions and utilizes cryptographic hashing to ensure the integrity and authenticity of the documents. In our proposed

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

architecture, users can upload their documents to an encrypted off-chain storage system such as the Inter Planetary File System (IPFS), while the corresponding hash of each document is stored immutably on the blockchain. Any tampering or alteration of the document changes the hash, which enables instant verification and detection of any unauthorized changes. Furthermore, access to documents is managed through smart contracts, which define who can access which document and under what conditions, offering a fine-grained access control mechanism [2]. One of the standout advantages of this system is its transparency and auditability. Every action, from document uploads to permission changes and verifications, is recorded on the blockchain. This provides a clear and immutable audit trail, increasing trust among users and organizations. It is especially beneficial in legal, academic, and administrative contexts where document authenticity and traceability are critical. Moreover, the system integrates user authentication through decentralized wallets, such as MetaMask, thereby removing the need for username-password systems and enabling a more secure, blockchain-native identity framework. This aligns well with the emerging trend of self-sovereign identity (SSI), which advocates for giving individuals control over their digital identity and personal data. From a societal perspective, the potential applications of a blockchain-based digital locker are vast and impactful. In the field of education, institutions can issue tamper-proof diplomas and transcripts, which graduates can share with employers directly from their digital lockers. In healthcare, patients can maintain ownership over their medical records and share them securely with healthcare providers. In government services, citizens can store and present verified copies of identity cards, tax documents, and legal affidavits without the risk of forgery. Despite its promise, blockchain adoption is not without challenges. Issues such as scalability, transaction costs (especially on public chains like Ethereum), privacy concerns, and regulatory uncertainty must be addressed for mainstream deployment. To overcome some of these concerns, our system uses a hybrid architecture—combining off-chain storage for large document files and on-chain data for verification and access control metadata. This ensures scalability and cost efficiency while retaining the benefits of blockchain's integrity guarantees. The architecture also allows for interoperability with existing systems, meaning it can be gradually integrated into existing institutional workflows rather than requiring a complete overhaul. This flexibility makes it more viable for adoption by schools, hospitals, and governmental bodies already using centralized digital record systems. The significance of this project lies not only in its technical merit but also in its alignment with broader movements toward data democratization, digital sovereignty, and user-centric design [3]. As global societies increasingly emphasize privacy, security, and digital empowerment, blockchain-based digital lockers can become a foundational infrastructure enabling these goals. To summarize, this paper proposes a secure, user-controlled, and scalable Digital Locker system using blockchain technology that addresses the core limitations of existing document management solutions. Through cryptographic assurance, decentralized control, and verifiable transparency, the system empowers users to store, retrieve, and share their digital documents with complete trust and reliability. This introduction sets the stage for a deeper exploration of the system’s architecture, implementation methodology, security model, and practical applications, as discussed in the following sections of the paper [4].

2. Ease of Use

The Digital Locker is a blockchain-based platform designed to securely store, manage, and share digital documents using a decentralized framework. It ensures tamper-proof data integrity, decentralized control, and user-defined access permissions. By integrating blockchain with off-chain storage systems like IPFS, Digital Locker allows users to store digital assets in a verifiable and traceable manner without relying on centralized authorities. The core functionality is underpinned by smart contracts, which govern the logic for document ownership, access control, and transaction auditability [5]. The fundamental goal of Digital Locker using Block-chain is to provide a secure, user-centric solution for document management, where users retain complete control over their digital data. The platform allows

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

document owners to grant and revoke access to specific users by linking their wallet addresses to document identifiers through smart contracts. Each interaction whether upload, access, permission grant, or revocation is logged on the blockchain, providing a transparent audit trail for verification and compliance. The Digital Locker using Block-chain ecosystem consists of several interdependent modules[6]:

- **User Interface (UI)** for uploading, managing, and sharing documents.
- **Blockchain Layer** (e.g., Ethereum or Polygon) for handling ownership and access control through smart contracts.
- **Off-Chain Storage** using **IPFS** (InterPlanetary File System) to store encrypted document files.
- **Smart Contracts** that act as the logic layer for access verification, permissions, and logging.
- **API Gateway/REST Endpoints** for communication between the UI, blockchain, and IPFS.

This distributed architecture ensures that while document content is stored off-chain to maintain scalability, critical metadata such as document ownership, hash values, and access permissions is securely stored on-chain, thereby maintaining authenticity and integrity [7].

2.1.Roles and Responsibilities

The functionality and operation of the DD-Locker platform are distributed among different stakeholders and system components. Each role has specific responsibilities to ensure **1**the smooth operation of the system [8].

Document Owner: The Document Owner is any authenticated user who uploads documents to the system and retains primary control over those files.

Responsibilities

- Upload documents securely via the UI.
- Generate IPFS hashes for each document.
- Initiate and sign transactions to store metadata on the blockchain.
- Grant or revoke access to other wallet addresses.
- Monitor access logs and audit trail for document activities.
- Maintain their private key/wallet credentials securely (e.g., via MetaMask).

Document Viewer (Recipient/Verifier): A Viewer is any user with granted permission to view or download specific documents.

Responsibilities

- Use their wallet address to authenticate access requests.
- Send queries to the smart contract to validate access.
- View/download documents if access is verified.
- Ensure their MetaMask/Wallet Connect session is active for access attempts.

Smart Contract: Smart contracts operate on the blockchain and enforce the rules of document access, permission grants, revocations, and verification.

Responsibilities

- Store and manage document metadata (IPFS CID, owner address).
- Handle permission assignments and revocations.
- Validate document access rights during each access attempt.

2.2.Application Flow

The Digital Locker system supports a complete lifecycle for digital document management upload, store, permission management, verification, and auditing [9]. Below is a breakdown of the critical processes:

Document Upload

- User selects a file for upload via the UI.
- The document is encrypted and uploaded to IPFS.
- IPFS returns a Content Identifier (CID), a unique hash corresponding to the file.

Smart Contract Interaction: The user initiates a smart contract function call grant Access (address viewer, string document ID) with:

- Their wallet (owner) address (automatically included in tx).
- Viewer's wallet address.
- Document identifier or hash.

Blockchain Logging

- The transaction is recorded on the blockchain.
- Gas fee is deducted from the owner's wallet.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

- Event logs for permission grant are created for transparency [10].

Revoke Access: This allows the document owner to revoke previously granted access from any user.

- Owner invokes the revoke Access (address viewer, string document ID) function.
- The revocation is written as an immutable transaction on the blockchain.
- Event logs record the revocation with timestamp and affected address.

Validate Access: Before any document is viewed or downloaded, the system performs access validation to ensure the request is legitimate.

- The viewer initiates a document access request.
- If the response is true:
  - The IPFS CID is returned.
  - The document is decrypted and served for viewing or download.
- If the response is false:
  - Access is denied.
  - The user receives an error message explaining the failure.

2.3.Key Highlights of Implementation and Testing

End-to-End Testing of Document Lifecycle: Each functional module—upload, store, share, verify—is tested individually and in sequence. Automated tests are implemented using frameworks such as:

- **Truffle/Hardhat** for smart contract testing.
- **Mocha/Chai** for JavaScript-based integration tests.
- **Postman/Newman** for API endpoint validation.

Tests include:

- Document upload with valid/invalid files.
- Smart contract permission flows.
- User access attempts with and without rights.

Wallet-Based Permission Testing: Permissions are tested using wallet tools like MetaMask and Wallet Connect:

- Grant and revoke permissions across multiple accounts.
- Test authentication failures (wallet not connected, wrong network).
- Simulate front-end access attempts from unauthorized wallets.
- Perform wallet-switching scenarios to validate real-time permission changes.

API Integration Testing: The REST API is responsible for communication between UI and blockchain. Integration tests validate:

- Correct payload formation for smart contract calls.
- Accurate retrieval of access validation responses.
- Real-time synchronization between user actions and blockchain logs.
- Secure fetching and rendering of IPFS resources only after access is verified.

Error Handling and Fail-Safes: Robust error handling is implemented across modules:

- Blockchain failures (e.g., insufficient gas, revert conditions).
- IPFS upload errors or hash mismatch detection.
- Unauthorized access attempts with descriptive error messages.
- Front-end UI feedback for network connection issues, transaction failures, or invalid document hashes.

3. System Design

3.1.Authentication Module

This module is responsible for onboarding residents and requesters. Before running the application, the user has to have the MetaMask extension installed in the browser. Also, the user needs to log in to the MetaMask and select the network for ethers transactions. After successfully logging in, MetaMask provides a user address to the client. The application starts with sending a token generated by the server to the client. This token is used to sign the message. Users are asked to sign the token message which then the signature is verified by the server for validation. Once the signature is validated, the user is successfully authenticated to the system. The JSON Web Token (JWT) is stored in the session with an expiration of 120 minutes; once the token expires, the user is logged out from the system. Upon logging into the system (smart contract), the application checks if the user is already registered or not. If not, it is redirected to the registration page; else to the dashboard page. On the registration page, there are two different registration forms for requester and

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

resident, respectively.

Resident Authentication Module: During registration, the user is asked for basic info and a master key. The master key is used to generate a long derived key which is then used to encrypt different documents residents wish to upload. Each document is encrypted with a different slice of the derived key, and these slices are generated using the doc index of the documents. Encrypting different documents with different keys ensures the confidentiality of each document uploaded by the residents. The derived keys are generated using PBKDF2[5], which is a password-based key derivation function. It applies a pseudo-random function HMAC (Hash-based message authentication code). The master key has been hashed, and this hash is updated with the user address and secret key of the server. This hash is generated using SHA256 and stored as an access key in user Info in Smart Contract. This access key is used to validate the user while uploading, verifying, and sharing the document. Upon successful registration, a welcome mail is sent to the newly registered user.

Requester Authentication Module: During requester registration, the user is asked for basic info and a unique code. This Unique code is required to ensure that the requester is a well-known organization apart from the normal user or resident. The administrator of the application makes this identification. Once the requester submits the request, the application will generate the private, public key pair to the requester upon successful registration. This key pair is used to encrypt/decrypt the symmetric key of a document. The public key is stored in the smart contract while the private key is sent to the requester user via Mail. The user has to provide the secret key while downloading the document.

3.2.Document Upload and Grant Permission Module

Once the resident is logged in and registered successfully, it is now eligible to upload the document on the application. It is assumed that the personal document is signed with the issuer's private key. So that when the requester downloads the document, it will decrypt the document using the issuer's public key. In this way, the requester will be able to avoid any forgery related to the personal documents of residents. In case if we include the issuer as a privilege who is responsible for issuing the document to the resident, then we can use the concept of digital signature to maintain the authenticity of the document owned by the resident.

Document Upload by Resident: While uploading the document, the resident is required to enter the master key. This master key helps to generate the derived key for the encryption of the selected document. The master key is verified by creating its hash and comparing it with the hash already stored in the blockchain (at the time of resident registration). It ensures that the resident is authorized and authenticated correctly. The encryption of the uploaded document is done using symmetric key cryptography such as Advanced Encryption Standard(AES), which provides confidentiality to the documents uploaded by residents over the server. This encrypted document is then uploaded to the file server. The files are stored on the server in such a way that no one can reveal the uploaded document information such as document name, owner details, etc. The application uses Dropbox as the main file server. However, this can be changed to any file server or any distributed file server. This file server remains transparent to the user.

Grant Permission by Resident: Read permission can be granted by residents in two ways. One way is the residence initiated, and the other is requester initiated. In case of residence initiated, residents directly provide permission for their owned document using its master key and the requesters' address or email id. In the case of requester initiated, a requester can search for the documents based on the user address/user email of resident or document id of the document. The search result includes the document id and document name. Now requester can select a document and raise the request for access to the corresponding resident. The corresponding request mail is then sent to the owner of the document. Upon receiving the mail request, residents verify the request based on document id and resident relationship using the logs stored on the smart contract while uploading the document. Another verification on the requester address is done using the logs stored on the smart contract while requester registration. Suppose the request is correctly

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

validated and verified. In that case, the residents' master key is verified to ensure the authenticity of the resident as done with the access key (master key hash digest) of a resident user stored on a smart contract.

4. Implementation Details

The implementation of the Digital Locker system leverages a decentralized architecture using blockchain and distributed storage systems. The backend is developed using Python with the Flask microframe work, which provides a lightweight yet scalable platform to expose RESTful APIs for various document management functionalities. The backend interfaces with blockchain smart contracts, handles user authentication, manages access control, and connects with IPFS for off-chain storage. The design is modular and decouples business logic from storage and blockchain interaction layers. To facilitate cross-origin requests from front-end clients, Flask-CORS is used. This is particularly essential when the application is deployed with separate domains for the frontend (e.g., a React/Vue.js frontend) and the backend. CORS headers ensure that browser-based security restrictions do not block legitimate requests from authenticated users interacting with their digital documents. For communication with the Ethereum blockchain, the project uses Ether.py, a Python wrapper around the Ethereum JSON-RPC API. This library enables seamless interaction with smart contracts deployed on EVM-compatible blockchains. It allows for reading from and writing to smart contracts, fetching logs, and monitoring events. Users can perform actions such as uploading a document (storing its IPFS hash on-chain), granting access to another user (via wallet address), or revoking permissions, all through this module. Document storage is handled via IPFS (InterPlanetary File System) using Pinata, a managed IPFS gateway service that provides reliable pinning and access control. Documents are first encrypted using Python's cryptography module or the hashlib library, ensuring that even if an IPFS CID is publicly accessible, unauthorized users cannot view the file content. A combination of symmetric key encryption (for the document) and asymmetric encryption (for key sharing) is employed to maximize security and maintain confidentiality. Authentication and access control are implemented using JWT (JSON Web Tokens). Each user logs in using a wallet address (e.g., via MetaMask), which is cryptographically verified through message signing. Once the wallet signature is validated, the server issues a time-bound JWT token that encapsulates the user identity and permissions. This token is then used for all subsequent API requests, reducing the need for frequent blockchain verification and improving performance (Figure 1 to 4). The metadata for user accounts, document ownership, and non-sensitive audit logs is managed in Supabase, an open-source Firebase alternative based on PostgreSQL. Supabase is used for:

- Managing user wallet mappings
- Tracking document IDs and IPFS CIDs
- Storing access logs in parallel to on-chain logs (for faster lookup)
- Managing expired or revoked tokens

By combining blockchain immutability with Supabase's fast relational querying, the system achieves a balanced hybrid architecture that is both robust and performant. Security is a top priority in the system's design. All incoming requests are validated for JWT integrity, wallet address authenticity, and smart contract state. Error handling is enforced across all modules, including failed blockchain transactions, expired JWT tokens, invalid file hashes, and network failures with Pinata or Supabase. The Digital Locker system uses a powerful combination of technologies Flask for backend services, Ether.py for blockchain interactions, Pinata/IPFS for decentralized storage, cryptography/hashlib for encryption, JWT with signature verification for user authentication, and Supabase for metadata management. This modern, scalable, and secure architecture offers users a tamper-proof and transparent platform to manage their digital documents confidently and efficiently.

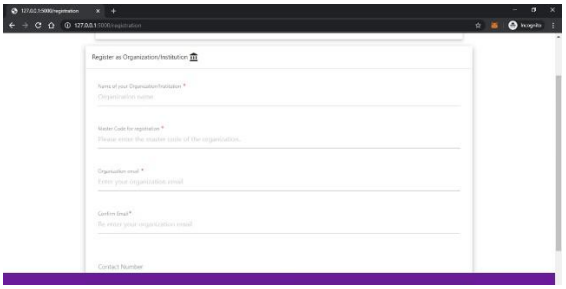

Figure 1 Registration

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

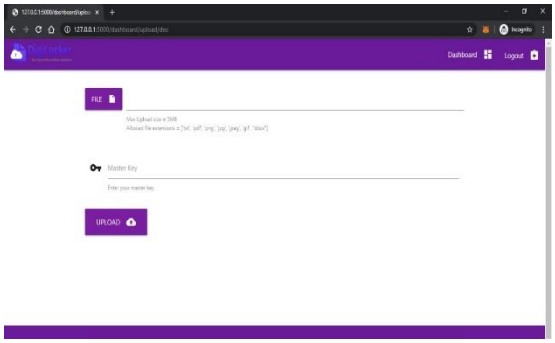

Figure 2 Upload file

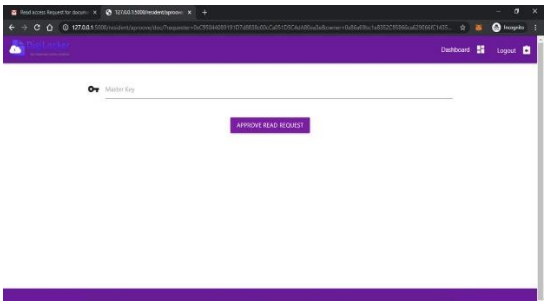

Figure 3 Master Key

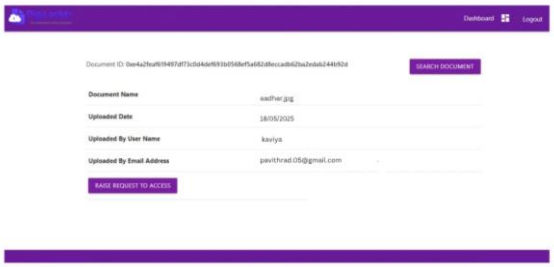

Figure 4 Search

Conclusion

In an increasingly digital world, the protection and management of sensitive documents have become more crucial than ever. The conventional methods of document storage primarily centralized cloud solutions are prone to manipulation, unauthorized access, data breaches, and lack of auditability. The Digital Locker using Blockchain addresses these concerns by introducing a decentralized, secure, and user-controlled platform for digital document management. This paper has presented the conceptualization, design, and implementation of DD-Locker, a blockchain-based digital locker system that enables individuals and organizations to store, verify, and share documents without relying on centralized authorities. The system leverages the core features of blockchain immutability, transparency, cryptographic assurance, and decentralized consensus to ensure that documents remain tamper-proof and access rights are always respected. The integration of IPFS allows the system to store large document files off-chain while preserving the integrity and authenticity of the data through on-chain hash storage. By encrypting documents client-side and linking their hash and CID to a smart contract, the system ensures that even if someone gains access to the CID, they cannot decrypt the file without authorization. This hybrid approach balances scalability with security, a vital feature for real-world applicability. The implementation also features smart contracts that govern the entire access control mechanism. From granting and revoking permissions to verifying access during document retrieval, every action is recorded on-chain, creating an immutable and transparent audit trail. This ensures that document access is both traceable and verifiable traits highly desirable in legal, academic, and institutional contexts. A significant strength of this project is its user-centric architecture. The blockchain-native identity system also aligns with the emerging concept of Self-Sovereign Identity (SSI), where users own and control their digital presence. Moreover, the system is designed to be modular and extensible. New features such as automated expiration of access rights, multi-signature access approval, or blockchain notarization of documents can be seamlessly integrated into the existing framework. However, there are certain limitations and challenges associated with this approach. One of the primary concerns is the transaction cost and speed on public blockchains. Additionally, privacy concerns such as revealing metadata like transaction timestamps or wallet addresses must be mitigated through techniques like zero-knowledge proofs or privacy-preserving layers. In conclusion, the Digital Locker using Blockchain stands as a compelling solution to the limitations of current digital storage systems. It enables secure, verifiable, and decentralized document management, placing ownership and control back in the hands of users. As society continues to demand more secure and trustworthy digital services, platforms like DD-Locker pave the way toward a future where digital trust is not assumed but cryptographically

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 06 June 2025
Page No: 2709 - 2716
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0402

guaranteed. Future work can involve deploying the system on Layer-2 scaling solutions like Optimism or zkSync, enhancing document privacy with Homomorphic Encryption, and integrating AI-based anomaly detection for monitoring unauthorized access patterns.

References

[1]. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
[2]. Wood, G. (2014). Ethereum: A Secure Decentralized Generalized Transaction Ledger.
[3]. Benet, J. (2015). IPFS - Content Addressed, Versioned, P2P File System. arXiv:1407.3561
[4]. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE Access.
[5]. Zyskind, G., Nathan, O., & Pentland, A. (2015). Decentralizing Privacy: Using Blockchain to Protect Personal Data. IEEE SPW.
[6]. Buterin, V. (2015). A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum White Paper.
[7]. Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K. L. (2017). BLOCKBENCH: A Framework for Analyzing Private Blockchains. SIGMOD.
[8]. Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., & Wan, J. (2018). Smart Contract-Based Access Control for the Internet of Things. IEEE IoT Journal.
[9]. Aitzhan, N. Z., & Svetinovic, D. (2016). Security and Privacy in Decentralized Energy Trading through Multi-Signatures, Blockchain and Anonymous Messaging Streams. IEEE Transactions on Dependable and Secure Computing.
[10]. ISO/TC 307. (2021). Blockchain and Distributed Ledger Technologies — Reference Architecture. International Organization for Standardization.