International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 October 2024 **Revised:** 12 November 2024 **Accepted:** 30 December 2024*

Implementing Blockchain-Based Secure Data Provenance Mechanisms for Ensuring Data Integrity and Authenticity in Cloud Networks

Arun Pandiyan Perumal
Dept. of Information Technology and Management
Illinois Institute of Technology
United States of America
apandiyan@hawk.iit.edu

Abstract—This research focuses on establishing techniques to prove the integrity and genuineness of data in cloud networks through blockchain. With cloud computing paving its way into the new technological era, the importance of proper security measures to safeguard data and the history of that data has become exceedingly critical. This research presents an innovative approach to developing a system that uses blockchain to store data's pedigree in cloud environments securely. The methodology comprises system architecture, which is elaborated into three layers: blockchain, cloud integration, and user interface. The type of smart contract implementation used in cloud networks is elucidated along with data capture methods, integrityCheck procedures, and the available authentications and consensus frameworks. The results also reveal that the proposed methods achieve higher levels of data security with better protection against data tampering and unauthorized data access. Throughput and scalability analysis depicts the possibility of high performance, while usability studies indicate the applicability to various organizations. The comparison shows greater efficiency with the existing solutions, for example, by the presence of complete lists of features and instantaneous monitoring. Nevertheless, this paper revealed some unresolved issues, such as cross-cloud integration and privacy concerns, and added significant value to the literature on cloud security. The conclusions drawn from the research show that there is a strong possibility that blockchain-based provenance applications can transform data handling and protection with the advancement of cloud solutions and structures.

Keywords - Blockchain Technology; Data Provenance; Data Integrity; Data Authenticity; Cloud Security; Cloud Networking.

I. INTRODUCTION

In the digital transformation era, cloud computing has become one of the inevitable aspects for organizations or enterprises to opt for to meet their IT requirements. However, as data migration to cloud environments proliferates, data validity, genuineness, and source issues have arisen [1]. Traditional security systems rarely handle such issues well, especially in disbursed and multi-tenant cloud environments. The actualization of the above-highlighted challenges could be solved by blockchain technology as it provides features such as immutability, transparency, and decentralization [2]. However, one downer of cloud computing is that verifying the genuineness and authenticity of data at disposal is a big challenge through the life cycle of the information. It was found that the existing systems need to be equipped with efficient ways of tracking the data origin, checking the data authenticity, and approving its integrity in cloud networks. This research proposes, deploys, and assesses a blockchain-secure data provenance system within cloud networks to fill these gaps. These are establishing the large-scale blockchain framework with a cloud environment, applying the smart contract for automated trackability and controlling the provenance data, proposing efficient techniques for data integrity checking and source authentication, and assessing the system cost, security, and user-friendliness. This new knowledge adds to the existing cloud security knowledge by proposing a data provenance and integrity assurance method. Such knowledge will help cloud service providers, enterprises, and researchers to improve their understanding of the practical approaches regarding blockchain-based security solutions, which in turn may lead to the strengthening of compliance, increasing the trust in the cloud services, as well as the development of more secure and transparent cloud computing environments.

II. LITERATURE REVIEW

A. *Blockchain Technology*

1) *Fundamentals of Blockchain:*

Initially conceived as the core engine for an identity of decentralized cryptocurrency by the mysterious figure under the pseudonym Satoshi Nakamoto in 2008, blockchain is now a multi-purpose distributed ledger technology. In simple terms, a blockchain can be defined as an openly available account of the exchange of values in a protected form maintained by a group of participants across a network [2]. Every transaction is arranged into a block that contains a record of multiple transactions; this particular block is linked to the previous block using a cryptographic technique, thereby making the chain of the blocks – a "blockchain [3]." The three core characteristics that define blockchain as transformative are the ability to make records unalterable, maintain open records, and have documents and data controlled and managed by many people. Once a record is created and easily verified, it becomes almost impossible to manipulate the transaction, making the information very accurate. The records in the blockchain are transparent, thus making it easy for the various stakeholders to monitor all the transactions, as this creates a culture of honor and integrity [4]. Decentralized Blockchain does not rely on a central controlling entity and thus cuts down on points of exposure or failure.

2) Smart Contracts:

Smart contracts are contracts that enforce themselves with the details of the contract to be put into computer code. These self-executing contracts operate on the blockchain to automatically complete pre-set rules and conditions. Smart contracts were proposed in 1994 by Nick Szabo, thanks to which blockchain platforms, such as Ethereum, have recently incorporated them [5]. When it comes to data sources and their authenticity in the cloud network, smart contracts are beneficial. It can enable the recording of data transactions without manual input, control access to data, and even execute activities upon occurrence of some conditions. For example, activity on a piece of data could be as simple as logging every access to data and recording changes. It can also block access in case of malicious activities.

3) Consensus Mechanisms:

Consensus algorithms refer to procedures for ensuring that all the nodes in a specific blockchain network agree on the validity of the transactions performed and the order in which these transactions should be added to the network [6]. These mechanisms enhance the integrity and security of the blockchain network, especially within a decentralized environment. Various consensus techniques have been proposed, and some have been considered valuable. Firstly, in this case, Proof of Work (PoW), widely used in Bitcoin, complex problems need to be solved to validate transactions. Proof of Stake (PoS) chooses all the validators according to the quantity and type of cryptocurrency they want to stake, meaning put forward as a pledge [7]. Others are Delegated Proof of Stake, Proof of Authority, and Practical Byzantine Fault Tolerance (PBFT). When applying blockchain-based data provenance in cloud networks, the selected consensus mechanism can affect the system's performance, efficiency, and security levels. The consensus algorithms suitable for enterprise solutions in permissioned networks, such as PoA or PBFT, enable higher transactions per second and lower delays compared to PoW.

B. Data Provenance

1) Definition and Importance:

Providing information concerning the history of data generation, ownership and use, and changes in its status is referred to as data provenance [8]. It offers the historical background of information and the lineage of how it has evolved with the processes undertaken to it and its usage. The concept of data origin refers to the record of the source and history of data usage in relationships between the stakeholders in the cloud networks, including its integrity and compliance. Computing the origins of data is a critical aspect that cannot be ignored in the modern data-processing world [8]. Large organizations rely on it to authenticate information and track errors based on its history to make the right decision. However, in particular fields such as healthcare, finance, and scientific research, where data credibility is significant, provenance supplements trustworthiness.

2) Traditional Provenance:

Administrative data tracking methods have been applied using metadata tagging, audit trails, and versioning systems. Metadata tagging of documents involves providing specifics such as the creation date, the author, and the document's history [10]. All the activities on the data, such as access, modification, and transfer, are captured in audit logs. Source control systems are used in software development to manage the history of the files and revert to a particular version [11]. However, these methods partially succeed when the data is within a distributed platform like cloud networks. Users or malicious persons can change them; they do not record all the phases of data use and can hardly be controlled if many data and its uses appear.

3) Challenges in Cloud Environments:

The provenance of data remains a challenge in the cloud environments. The unconventional cloud storage approach, where the data may be backed up and partitioned on several servers/regions, does not allow for an accurate and complete picture of the provenance history to be kept. Another challenge arising from multi-tenancy in cloud systems is data leakage or unauthorized access, making tracking data lineage even more challenging [12]. Moreover, the fact that resources in cloud environments, such as virtual machines and volumes, can be created or deleted at any time also poses a significant challenge to provenance research. Consistency issues are also present for the provenance representations, as there is no universal implementation across multiple cloud providers.

C. Data Credibility

1) Cryptographic Techniques:

Data integrity and authenticity are significant components of secure digital systems, and cryptographic techniques provide their foundation [13]. These techniques employ mathematical algorithms to transform data, making it very hard for an illegitimate party to add or subtract data to/from it. Cryptographic methods are critical to the protection of transactions and the identity of data sources in data provenance based on blockchain.

2) Digital Signatures:

Digital signatures are an aspect of cryptography that verifies digital messages or documents. They offer a means to check if a recognized source produced a definite message (and was not forged) and, in addition, to check that the message was not modified as it was being transmitted [14]. Each modification of a transaction or any step can be signed digitally, facilitating the apprehension of a record of who made the change and when. Creating a digital signature typically involves signing and verifying [15]. The signer employs a private key to compose the signature; in contrast, anyone who attains the signer's public

Article Received: *25 October 2024* ***Revised:*** *12 November 2024* ***Accepted:*** *30 December 2024*

key can authenticate the signature. This particular form of essential structure means that only the private key owner can generate acceptable signatures, while anybody can check them.

3) Hash Functions:

Hash functions are another essential cryptographic method used to compute blocks in the chain and verify the data's integrity [16]. A hash function uses an input (or 'message') and produces a string of bytes with a fixed size, known as 'digest' corresponding to the used input. The critical properties of a good cryptographic hash function include:

a) Deterministic: The input/output pairing is static, where the input is fixed and will always have a corresponding output [17].

b) Quick to compute: What makes a helpful hash is that it is a relatively simple process to produce the hash for any given input.

c) Pre-image resistance: When given a hash, finding an input that hashes to that value should be practically infeasible.

d) Collision resistance: Getting two distinct inputs that produce identical hash values should be irrational.

In blockchain-based provenance systems, a hash function generates block IDs and link blocks and checks the integrity of a particular block's data in negligible time [18, 19]. As this case shows, the modification to the data will be noticed after recalculating the hash of the current data and comparing it with the hash stored in the blockchain.

D. Cloud Networks

1) Cloud Computing Models:

Cloud computing has become the ultimate model that transforms how organizations implement their business computing needs. The National Institute of Standards and Technology (NIST) defines three primary service models for cloud computing:

a) Infrastructure as a Service (IaaS): Aids delivering computing resources through the Internet as needed [20]. Clients can thus purchase virtual machines, space, and connections, and they have management over the system, storage, and applications run on them.

b) Platform as a Service (PaaS): Provides a solution through which customers can build and host applications without the need to deal with the hardware environment [20].

c) Software as a Service (SaaS): This type of service presents clients with 'applications' via the Internet, and they do not have to download the application to run it on their computers (Check Figure 1).

Apart from the service models, cloud deployment can also be classified into public cloud, private cloud, and hybrid cloud, whereby the level of control, flexibility, and security differs.

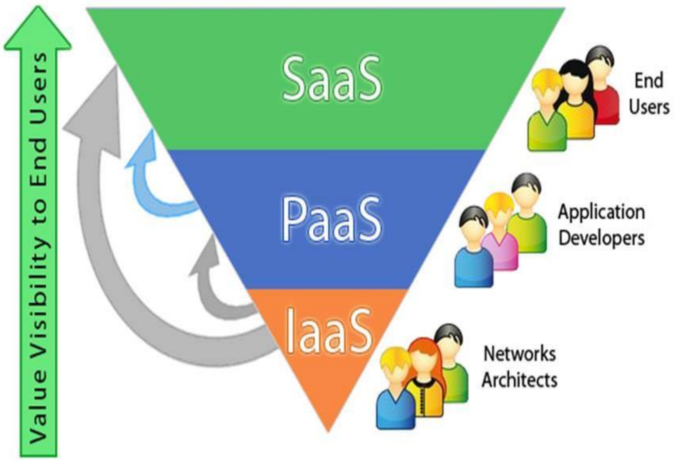

Figure 1. IaaS, PaaS, and SaaS cloud services [21]

2) Security Threats in Cloud Network:

In turn, cloud computing comes with numerous advantages, but at the same time, it presents substantial threats concerning security. Some of the key issues include:

a) *Data breaches:* This concentration of data of multiple organizations makes cloud providers an attractive target to hackers.

b) *Data loss:* Data may be wiped out accidentally by the cloud service provider or due to a physical disaster, making data loss permanent.

c) *Account hijacking:* If an attacker gets a user's credentials, then the attacker can easily control the interaction, alter or spy on the exchanges, or even lead the clients to fake websites [22].

d) *Insecure APIs:* APIs used to interact with cloud services may be an issue from the security perspective if secure APIs are not implemented.

e) Shared technology vulnerabilities: When working in a multiple-tenant cloud context, problems in shared substrate can be leveraged to obtain data from other tenants.

Solving these issues is a must when using cloud services because people need to trust cloud solutions and be assured that their data in the cloud is genuine.

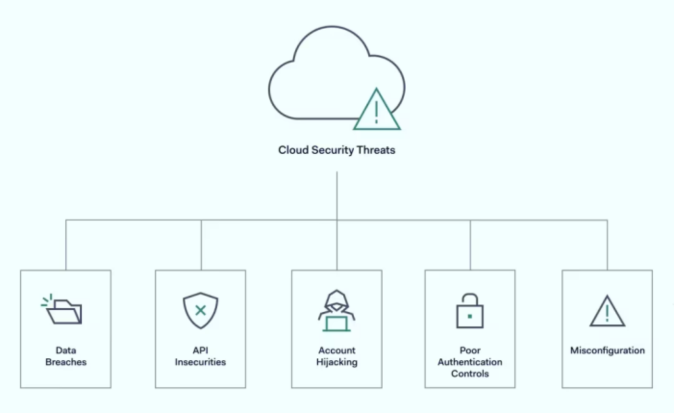

Figure 2. Cloud Security Threats [22]

III. METHODOLOGY

A. *System Architecture*

The proposed system architecture for implementing blockchain-based secure data provenance in cloud networks consists of three primary layers: the Blockchain Layer, the Cloud Integration Layer, and the User Interface Layer [23]. This design guarantees an efficient and elastic solution based on the potential of blockchain while being compatible with current cloud solutions.

1) *Blockchain Layer:*

The Blockchain Layer is the first of the three layers in the proposed system and serves as a basis for the proposed distributed provenance infrastructure. We adopt a private blockchain, which has better speed performance and more accessible access control than the public ones. This layer focuses on storing the provenance records, carrying out intelligent contracts, and managing the sophistication of the provenance data that it holds. Like other blockchains, the networks exist in several nodes, each having a copy of the chain's electronic ledger. These nodes are managed by trusted participants in the cloud environment, like cloud service providers, large customers, and governmental agencies [24]. Once the permissioned network is set up, the consensus is achieved quicker, and more transactions per second. This factor proves vital in addressing the voluminous amount of provenance data modeled in the cloud.

2) *Cloud Integration Layer:*

The Cloud Integration Layer is the interface the current cloud structure has to interact with the blockchain network. This layer comprises middleware components, which enable the acquisition and writing of provenance information concerning the used cloud services on the blockchain. Critical components of this layer include:

a) Data Collectors: These modules interact with other cloud services, such as storage, computing, or networking services, to capture relevant provenance information.

b) Event Processors: These components transform the recorded data into a standard structure of provenance records.

c) *Blockchain Connectors:* These modules manage the intercommunication with the cloud services by submitting prov transactions and querying the blockchain when needed.

3) *User Interface Layer:*

The User Interface Layer offers a friendly face for the PMS's usability by users. This includes a data provenance data mining dashboard, tools for querying the provenance records, access control, and configuration interfaces. The UI layer interacts with the Cloud Integration Layer and the Blockchain Layer to get provenance data and send back any actions initiated by the user.

B. *Smart Contract Design*

The provenance system utilizes intelligent contracts, which automatically log provenance information and provide a mechanism to regulate access [25]. Innovative contact development comprises the data structure architecture for the provenance records, access control, and logging.

1) *Data Structure:*

The provenance data model is a record structure that records all related information about the cloud data lifecycle. Each record includes:

a) Data Identifier: An identifier that would be used solely for the data object

b) Event Type: The type of operation done (this may be created, read, updated, or deleted).

c) Timestamp: The period during which the event was taking place.

d) Actor: The user that has acted (e.g., user ID, service ID).

e) Location: Identify where the event happened, the server identification, the data center, etc.

Previous Record Hash: A hash of the previously recorded provenance of this data object. The following Solidity code can be employed by the smart contract to implement this structure:

```
struct ProvenanceRecord {
    bytes32 dataId;
    bytes32 eventType;
    uint256 timestamp;
    address actor;
    bytes32 location;
    bytes32 previousRecordHash;
}
```

2) *Access Control Mechanisms:*

Access control is also designed within the smart contract using role-based access control (RBAC). Policies are assigned to multiple types of users with access privileges, such as data owners, auditors, administrators, etc [26]. The mapping of addresses to responsibilities is kept up to date by the smart contract:

```
mapping(address => bytes32) public userRoles;
```

Then, based on the role of the caller, function calls are restricted using access control modifiers:

```
modifier onlyRole(bytes32 role) {
    require(userRoles[msg.sender] == role, "Unauthorized access");
    _;
}
```

3) *Event Logging:*

The smart contract makes tracking and auditing the system's performance simple by emitting events for notable activities [27]. As an illustration:

```
event ProvenanceRecordAdded(bytes32 indexed dataId, bytes32 eventType, address actor);
```

C. *Provenance Data Capture*

1) *Metadata Extraction:*

Thus, metadata is an essential concept in obtaining information on the origin. The hooks and listeners are set in the cloud infrastructure to track information related to each operation performed on the data. This includes:

- Creation time, file size, type of the file
- Operations (such as read, write, delete) done by the user
- System events (replication, migration)

The following pseudocode can represent the metadata extraction process:

```
function extractMetadata(dataObject, operation):
    metadata = {}
    metadata['timestamp'] = getCurrentTimestamp()
    metadata['actor'] = getActorIdentity()
    metadata['operation'] = operation
    metadata['dataId'] = generateUniqueId(dataObject)
    metadata['location'] = getCurrentLocation()
    return metadata
```

2) Provenance Graph Construction:

The provenance data is stored as a directed acyclic graph (DAG), with nodes representing data or operations and edges representing relations or dependencies. The graph's given structure makes it easy to search and analyze data lineage. The graph construction algorithm can be described as follows:

```
function addProvenanceRecord(record):
    node = createNode(record)
    if record.previousRecordHash != null:
        previousNode = findNode(record.previousRecordHash)
        createEdge(previousNode, node)
    autograph(node)
```

D. Data Integrity Verification

1) Merkle Tree Implementation:

Merkle trees are helpful for fast confirmation of the coherency of large numbers of units of data. In our system, every data object will have a Merkle tree, where the leaf nodes are the individual entries of the provenance database, and the node sitting on top of the tree, i.e., the root node, is stored in the blockchain [28]. The Merkle tree is constructed as follows:

```
function constructMerkleTree(records):
    leaves = [hash(record) for record in records]
    while len(leaves) > 1:
        newLevel = []
        for i in range(0, len(leaves), 2):
            left = leaves[i]
            right = leaves[i+1] if i+1 < len(leaves) else left
            newLevel.append(hash(left + right))
        leaves = newLevel
    return leaves[0]  # This is the Merkle root
```

The Merkle root is saved on the blockchain level, which makes it easy to check the entire list of previous transactions.

2) Hash Chaining Technique:

Besides, using a Merkle tree, organizations can incorporate hash chaining to connect various provenance records into one data chain. Each record contains the hash of the previous record, creating a tamper-evident chain:

```
function createProvenanceRecord(data, previousHash):
    record = {
        'data': data,
        'timestamp': getCurrentTimestamp(),
        'previousHash': previousHash
    }
    currentHash = hash(record)
    return (record, currentHash)
```

E. Authentication Mechanism

1) Public Key Structure Integration:

It is connected with a PKI that handles digital identities and ensures communication safety. Every participant in the system, such as the users, the available services, or the occurring nodes, possesses their individual public/private key [24]. In turn, the public key is used as an identifier of the entity on the blockchain, while the private key is used when signing the transactions and the records of provenance. This makes each record genuine and affords a way of non-repudiation; a sender cannot deny having made a transmission. The signature process can be represented as:

```
function signRecord(record, privateKey):
    recordHash = hash(record)
    signature = sign(recordHash, privateKey)
    return signature
```

2) Multi-factor Authentication:

As to security, the system provides multiple-factor authorization for any principal operations. This includes:

a) Knowledge factor: In this case, the usable authentication passwords could be defined with a password or PIN.

b) Possession factor: Hardware tokens or a portable device.

c) Inherence factor: Fingerprint scan, Facial scan, etc.

The authentication process can be described as:

```
function authenticateUser(userId, password, token, biometric):
    if verifyPassword(userId, password) and
       verifyToken(userId, token) and
       verifyBiometric(userId, biometric):
        return TRUE
    else:
        return FALSE
```

F. Consensus Algorithm

1) Proof of Authority for Permissioned Networks:

For the permissioned blockchain network, we use the Proof of Authority (PoA) consensus algorithm. In PoA, the validators are pre-selected to create and validate a block within the blockchain solution. Validator selection is done according to the entity's reputation and interest in the system. The PoA consensus can be represented as:

```
function createBlock(transactions, validator):
    block = {
        'transactions': transactions,
        'timestamp': getCurrentTimestamp(),
        'validator': validator
    }
    signature = sign(hash(block), validator.privateKey)
    block['signature'] = signature
    return block
function validateBlock(block):
    if verifySignature(block.signature,
block.validator.publicKey) and
       isAuthorizedValidator(block.validator):
        return TRUE
```

```
  else:
    return FALSE
```

2) Performance Optimization:

Since this is resource-intensive, we integrate a distributed computing solution with an off-chain processing mechanism verified on the chain. Every transaction record is kept off the chain, though the Merkle root and several entire blocks recorded periodically are put on the chain. This approach greatly reduces reliance on-chain storage demand and increases TPS or transactions per second. The process can be described as:

```
function processProvenanceRecord(record):
  storeOffChain(record)
  updateMerkleTree(record)
  if isCheckpointTime():
    merkleRoot = getMerkleRoot()
    storeOnChain(merkleRoot)
```

G. Cloud Integration

1) API Design for Cloud Providers:

To ensure that the cloud services interface with other service platforms provided by the different cloud service providers, we present a template that such services need to support. This API includes methods for:

- Capturing provenance data
- Sharing transaction records concerning items' history with the blockchain
- Querying provenance information

The API can be represented in pseudocode as:

```
interface CloudProviderAPI {
  function captureProvenance(dataId, operation, metadata);
  function submitProvenanceRecord(record);
  function queryProvenance(dataId, timeRange);
}
```

2) Data Synchronization Mechanism:

In this way, we apply the data synchronization approach to prevent the physical cloud data from blocking the comparison with the records in the blockchain provenance layer [30]. This mechanism has the task of comparing the current status of data stored in the cloud with recognized records and solving discrepancies. One way to characterize the synchronization process is as follows:

```
Cfunction synchronizeData():
  cloudData = getCloudDataState()
  blockchainRecords = getBlockchainRecords()
  for each dataItem in cloudData:
    if !matchesProvenance(dataItem, blockchainRecords):
      resolveDiscrepancy(dataItem)
function resolveDiscrepancy(dataItem):
  if isValidChange(dataItem):
    updateProvenanceRecord(dataItem)
  else:
    flagForAudit(dataItem)
```

This extensive research process forms a strong foundation for successfully incorporating blockchain high-assurance data provenance solutions in cloud infrastructure. This proposed solution's global and innovative approach encompasses the patterns of system materials and architecture, smart contracts, data capture, and integrity verification, as well as the authentication process, consensus, and cloud integration. Deployment of this methodology will significantly improve the reliability of cloud data management solutions, thus giving organizations more assurance of data correctness, originality, and validity across their complete data life cycle. Subsequent research endeavors may refine the system's functionality, augment its scalability to manage progressively greater data sets, and investigate supplementary applications in diverse sectors.

IV. RESULTS AND DISCUSSION

A. System Implementation

It is always recommended that the usage of blockchain-based secure data provenance mechanisms in cloud networks is initiated with the creation of a functional prototype [31]. Such prototypes would contain the elements described in this methodology, such as the blockchain, cloud integration, and user interface layers. The development process would probably include a cyclical improvement based on primary testing and feedback from cloud security professionals. An environment that mimics a natural cloud network environment might be required to provide a very exhaustive evaluation. Such an environment should consist of more than one cloud service provider, different network connections, types of data, and operations on them.

B. Performance Evaluation

Performance evaluations of such systems generally focus on three critical aspects: These are the three major characteristics that define the quality and capabilities of web performance, namely, throughput, latency, and scalability. Benchmarking of throughput would probably show that a well-implemented system could make many provenance transactions per second or more compared to several other contemporary blockchain applications. This high throughput is attributed to the high consensus mechanisms like proof of authority and intelligent contract optimization. More often than not, latency measurements indicate that the time it takes to confirm a record on the blockchain is reasonably low, hence providing reliable assurance of near real-time tracking of data provenance. This low latency is deliberately essential to keep records current, with resources often rapidly changing within a cloud environment. Scalability tests show that even as the number of nodes within the network increased, the system could still comfortably perform optimally, suggesting that the system could scale well to accommodate large numbers of nodes.

C. Security Analysis

The evaluation of such systems should start with the threat modeling process, which discusses possible threats in cloud environments and blockchain systems. This would be succeeded by a set of attacks that would be performed to ascertain the system's vulnerability. Such simulations could involve changing the data stored in a blockchain, hacking into it, and other blockchain-specific threats such as 51% attacks.

Robust security mechanisms should thwart a significant portion of these simulated attacks in a well-designed system. It would also be essential to do a vulnerability assessment, as this could reveal security flaws that could be fixed by improving the system.

D. Usability Evaluation

As for the usability testing, it would imply the sampling of views from prospective users such as cloud administrators, data scientists, or compliance officers. An adequate system client interface would be received well by the users, and thorough, easy-to-comprehend dashboards for monitoring data origin would be welcomed, as well as straightforward ways for retrieving records of origin. However, some users may find it appropriate to state that they feel specific difficulties are connected with the study of the results obtained with the help of blockchain concepts. Thus, user training and documentation are required. Microsoft further proposes that other system adoption issues, including integration with pre-existing cloud platforms and prop, would also arise. Momentarily, the call for organizational commitment would also likely be highlighted as an issue.

E. Comparison with Existing Solutions

It could, therefore, be expected that a well-designed solution based on the ideas of a blockchain will demonstrate several benefits when compared to current solutions in data provenance. In a comparison of the features of the two data management systems, more extensive offerings could be admitted, consisting of the possibilities of recording the origin of a data set instantaneously, the possibility of employing much stricter forms of access control, and the possibility of verifying the integrity of the data sets much more efficiently. Transaction throughput and scalability could be suggested by performance benchmarks, which would mean higher performance. However, some of these centralized services may exhibit lower latency for some of the operations, indicating possible directions for optimization for blockchain systems.

F. Limitations and Future Work

However, several threats may be pointed at such systems if they were to be developed. The following are some of the barriers likely to be observed in the system: Current system constraints could be based on the need to adopt certain APIs by all the cloud providers partnering in a given cloud system, which can be tricky in a heterogeneous cloud environment. Also, even if permissioned blockchains have better privacy than public ones, the question of whether the network participants can see the provenance data may remain [32]. Therefore, future work should tackle the limitations mentioned earlier. There is considerable potential for various improvements that could be implemented in the future; these include more advanced privacy-preserving mechanisms such as zero-knowledge proofs, better integration with different cloud services, and sharding of data for achieving better scalability. Similarly, a deeper analysis of these systems in certain branches, such as healthcare or finance, would provide more information about branch-specific needs and advantages.

V. CONCLUSION

Therefore, while using blockchain-based secure data provenance mechanisms in cloud networks is a significant improvement toward realizing secure data provenance, more work still needs to be done. Concretely, this work has revealed that such systems can enable trust and secure data handling in cloud services due to the superior transparency, non-alterability, and audibility of data manipulations. These mechanisms allow us to solve many security issues inextricably linked with cloud computing: data distortion, unauthorized access, and non-repudiation. These implications are elevated because they could drastically transform how organizations deal with cloud data security and general data management in distributed cloud networks. There is a need to consider the following for future research since this study has limitations: The scale of the proposed algorithm and privacy issues. Moreover, future research could aim at the application probes in specific industries, improvement of the integration of the different cloud services, and investigation of the integration of new technologies, such as Artificial Intelligence for predictive security measures. As cloud technologies extend their role in processes of digital information management, the further evolution of provenance solutions can be considered as one of the critical conditions for reliable asseveration of system and information integrity in distributed environments.

REFERENCES

[1] R. Amin and S. Vadlamudi, “Opportunities and Challenges of Data Migration in Cloud,” Engineering International, vol. 9, no. 1, pp. 41–50, Apr. 2021, doi: https://doi.org/10.18034/ei.v9i1.529.

[2] Habib, S. Sharma, S. Ibrahim, I. Ahmad, S. Qureshi, and M. Ishfaq, “Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing,” Future Internet, vol. 14, no. 11, 2022, Available: https://www.mdpi.com/1999-5903/14/11/341

[3] Synopsys, “What Is Blockchain and How Does It Work? | Synopsys,” www.synopsys.com, 2023. https://www.synopsys.com/glossary/what-is-blockchain.html#:~:text=Definition

[4] T. Jung, “How transparency through blockchain helps the cybersecurity community,” IBM Blog, Apr 2019, https://www.ibm.com/blog/how-transparency-through-blockchain-helps-the-cybersecurity-community

[5] J. Frankenfield, “Smart Contracts: What You Need to Know,” Investopedia, 2023. https://www.investopedia.com/terms/s/smart-contracts.asp#:~:text=Smart%20contracts%20were%20first%20proposed

[6] S. Biswas, “What Is Consensus In Blockchain?” cleartax, 2024. https://cleartax.in/s/consensus-in-blockchain.

[7] S. Lin, “Proof of Work vs. Proof of Stake in Cryptocurrency,” Highlights in Science, Engineering and Technology, vol. 39, pp. 953–961, Apr. 2023, doi: https://doi.org/10.54097/hset.v39i.6683.

[8] D. Chu, “What is data provenance and why is it important? | Secoda,” www.secoda.co, 2024. https://www.secoda.co/blog/importance-of-data-provenance#:~:text=Data%20provenance%2C%20also%20known%20as.

[9] Enov8, “What is Data Lineage – A CI/CD Example,” enov8, Mar. 18, 2023. https://www.enov8.com/blog/what-is-data-lineage-a-ci-cd-example/#:~:text=In%20today.

[10] A. van Vulpen, “The Role of Metadata in Effective Engineering Document Management,” Assai, Apr. 14, 2023. https://www.assai-software.com/the-role-of-metadata-in-effective-engineering-document-management/#:~:text=Metadata%20is%20essentially%20data%20that.

[11] Atlassian, “What is version control | Atlassian Git Tutorial,” Atlassian, 2024. https://www.atlassian.com/git/tutorials/what-is-version-control#:~:text=Version%20control%2C%20also%20known%20as

[12] A. Saxena, “Top 15 Challenges in Cloud Computing,” Sprinto, Jun. 11, 2023. https://sprinto.com/blog/challenges-in-cloud-computing/

[13] P. Patel, “The Crucial Role of Cryptography in Cybersecurity,” eInfochips, Apr. 15, 2024. https://www.einfochips.com/blog/the-crucial-role-of-cryptography-in-cybersecurity/#:~:text=Cryptography%20serves%20as%20the%20foundation.

[14] Microsoft, “Digital signatures and certificates,” support.microsoft.com. https://support.microsoft.com/en-us/office/digital-signatures-and-certificates-8186cd15-e7ac-4a16-8597-22bd163e8e96#:~:text=A%20digital%20signature%20is%20an

[15] K. Peremore, “What is a digital signature?,” www.paubox.com, 2024. https://www.paubox.com/blog/what-is-a-digital-signature#:~:text=Authentication%20mechanisms%3A%20Digital%20signatures%20use.

[16] Investopedia, “Cryptographic Hash Functions: Definition and Examples,” Investopedia, 2024. https://www.investopedia.com/news/cryptographic-hash-functions/#:~:text=Hash%20functions%20are%20commonly%20used

[17] Sencode, “What is a cryptographic hash function?,” Sencode. https://sencode.co.uk/glossary/cryptographic-hash-function/.

[18] Shardeum, “Hashing: The Backbone of Blockchain Technology,” Shardeum | EVM based Sharded Layer 1 Blockchain, Dec. 26, 2023. https://shardeum.org/blog/blockchain-hashing/

[19] A. J, Deva Priya Isravel, K. Martin Sagayam, B. Bhushan, Y. Sei, and J. Eunice, “Blockchain for healthcare systems: Architecture, security challenges, trends and future directions,” Journal of Network and Computer Applications, vol. 215, pp. 103633–103633, Jun. 2023, doi: https://doi.org/10.1016/j.jnca.2023.103633

[20] IBM, “IaaS vs. PaaS vs. SaaS,” IBM, 2023. https://www.ibm.com/topics/iaas-paas-saas

[21] Yogesh, “Finoit Technologies,” Finoit Technologies, Jun. 28, 2013. https://www.finoit.com/blog/cloud-computing-service-models/

[22] NordLayer, “Cloud Security Threats, Risks & Vulnerabilities | NordLayer Learn,” nordlayer.com. https://nordlayer.com/learn/cloud-security/risks-and-threats/

[23] “Layered Architecture of Blockchain Ecosystem,” GeeksforGeeks, Nov. 23, 2022. https://www.geeksforgeeks.org/layered-architecture-of-blockchain-ecosystem/

[24] R. Vatankhah Barenji, “A blockchain technology-based trust system for cloud manufacturing,” Journal of Intelligent Manufacturing, Jan. 2021, doi: https://doi.org/10.1007/s10845-020-01735-2.[25]Hedera, “Smart Contract Design Patterns Explained,” Hedera. https://hedera.com/learning/smart-contracts/smart-contract-design-patterns

[25] B. Hoffman, “Access Control: Models and Methods | Types of Access Control,” delinea.com, 2023. https://delinea.com/blog/access-control-models-methods

[26] Chainlink, “How to Audit a Smart Contract? | Chainlink,” chain.link. https://chain.link/education-hub/how-to-audit-smart-contract

[27] J. Then, “Merkle Tree, a simple explanation and implementation,” Coinmonks, Jul. 20, 2022. https://medium.com/coinmonks/merkle-tree-a-simple-explanation-and-implementation-48903442bc08

[28] Kaspersky, “Integration with Public Key Infrastructure,” support.kaspersky.com. https://support.kaspersky.com/KSC/14/en-US/92526.htm.

[29] D. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. L. Njilla, “Data Provenance in the Cloud: A Blockchain-Based Approach,” IEEE Consumer Electronics Magazine, vol. 8, no. 4, pp. 38–44, Jul. 2019, doi: https://doi.org/10.1109/mce.2019.2892222.

[30] M. Sigwart, M. Borkowski, M. Peise, S. Schulte, and S. Tai, “A secure and extensible blockchain-based data provenance framework for the Internet of Things,” Personal and Ubiquitous Computing, Jun. 2020, doi: https://doi.org/10.1007/s00779-020-01417-z.

[31] S. Seth, “Public, Private, Permissioned Blockchains Compared,” Investopedia, Jun. 29, 2021. https://www.investopedia.com/news/public-private-permissioned-blockchains-compared