ForensiBlock: A Provenance-Driven Blockchain Framework for Data Forensics and Auditability

Asma Jodeiri Akbarfam[1], Mahdieh Heidaripour[1], Hoda Maleki[1], Gokila Dorai[1], and Gagan Agrawal[2]

[1]*School of Computer and Cyber Sciences, Augusta University* Augusta, USA

ajodeiriakbarfam, mheidaripour, hmaleki, gdorai@augusta.edu

[2]*School of Computing, University of Georgia* Athens, USA gagrawal@uga.edu

Abstract—Maintaining accurate provenance records is paramount in digital forensics, as they underpin evidence credibility and integrity, addressing essential aspects like accountability and reproducibility. Blockchains have several properties that can address these requirements. Previous systems utilized public blockchains, i.e., treated blockchain as a black box, and benefiting from the immutability property. However, the blockchain was accessible to everyone, giving rise to security concerns and moreover, efficient extraction of provenance faces challenges due to the enormous scale and complexity of digital data. This necessitates a tailored blockchain design for digital forensics. Our solution, Forensiblock has a novel design that automates investigation steps, ensures secure data access, traces data origins, preserves records, and expedites provenance extraction. Forensiblock incorporates Role-Based Access Control with Staged Authorization (RBAC-SA) and a distributed Merkle root for case tracking. These features support authorized resource access with an efficient retrieval of provenance records. Particularly, comparing two methods for extracting provenance records – off-chain storage retrieval with Merkle root verification and a brute-force search – the off-chain method is significantly better, especially as the blockchain size and number of cases increase. We also found that our distributed Merkle root creation slightly increases smart contract processing time but significantly improves history access. Overall, we show that Forensiblock offers secure, efficient, and reliable handling of digital forensic data.

Index Terms—Blockchain, provenance, data forensics, security, access control, verification.

I. INTRODUCTION

Digital forensics is crucial in modern investigations, enabling law enforcement agencies and organizations to extract, analyze, and preserve digital evidence for legal proceedings [1], [2]. However, safeguarding evidence, particularly the digital evidence, throughout the entire investigative process remains a primary challenge [3]. The Chain of Custody (CoC) plays a vital role in maintaining the integrity and credibility of digital evidence in digital forensics. CoC involves meticulously recording every transaction related to digital forensic evidence and maintaining a comprehensive storage history since its creation [3]. Traditional evidence collection and preservation methods often struggle to provide comprehensive and transparent provenance records, leading to challenges in establishing trust and accountability [4].

A standardized approach to preserving CoC is essential, highlighting the significance of data provenance in digital forensics [5]. Data provenance is the ability to trace and authenticate any data artifact's origin, custody, and history. Particularly, provenance records play a critical role in ensuring the integrity and reliability of evidence and addressing issues of evidence tampering and data manipulation.

In recent years, blockchain technology has emerged as a promising solution for addressing data provenance needs across different applications, including digital forensics [6]. As an immutable and decentralized ledger, blockchain offers a tamper-proof and transparent framework for recording and verifying the flow of digital evidence [7]. By implementing a provenance-driven blockchain system, it becomes possible to establish a robust and auditable CoC for digital evidence, enhancing both privacy protection and data integrity.

Several approaches have been designed to integrate blockchain into digital forensics [8], [9]. These approaches predominantly utilize public blockchains such as Ethereum. However, they face notable drawbacks that limit their widespread adoption and effectiveness. For instance, employing Ethereum as a public blockchain raises concerns about data accessibility and confidentiality, potentially jeopardizing sensitive investigative information [10]. Furthermore, existing approaches in digital forensics on blockchains such as [1], [5], [10], [11] have shown promising advancements in the domain. However, these approaches lack the integration of the essential elements, which we argue are traceability, integrity [12], automated and secure authentication and access control, immutability, fast extraction of provenance records with verification, version tracking of data and secure communication between components. Particularly, in some cases, the limitations stem from the prevailing treatment of blockchain as a logging tool in digital forensics workflows, prioritizing alternative design aspects. Overall, there is a need for a framework based on a private blockchain that specifically addresses each of the above requirements of provenance in digital forensics. Such a design should solve privacy, security, and scalability challenges while preserving the desired benefits of transparency and immutability in digital forensics.

This paper presents ForensiBlock, a private blockchain solution specifically designed to overcome the limitations of existing blockchain systems in digital forensics provenance.

ForensiBlock incorporates various essential components to enhance the investigation process and ensure comprehensive data management. It enables tracking all data involved in an investigation, including communication records, while providing the ability to trace data back to its origin. ForensiBlock also facilitates fast extraction and verification of evidence, ensuring efficiency and accuracy throughout the investigation. ForensiBlock also utilizes cryptographic techniques to protect sensitive information. It includes access control mechanisms that detect and handle malicious access requests, ensuring that only authorized individuals have appropriate access to the data. The system also supports stage changes in the investigation, allowing for seamless progression and maintaining data integrity.

The main contributions of this paper are as follows:

- Identification of the significance of provenance in digital forensics and addressing the research gap in current blockchain-based digital forensics provenance research.
- Proposal of a novel access control method specifically designed to meet the access control needs in digital forensics, automating the investigation process.
- Presentation of a distributed Merkle tree for verifying the integrity of extracted cases.
- Design and development of Forensiblock, a specialized framework tailored for digital forensics provenance that preserves all relevant records.
- Implementation of extraction capabilities to enable timely retrieval of provenance records.
- Conducting experiments to evaluate the capabilities and performance of Forensiblock.

The rest of this paper is structured as follows. Section II discusses an overview of blockchain, provenance, and digital forensics. Section III explores the related work. Section IV describes the framework and the proposed protocols. Section V illustrates the implantation and the experiment, and the paper concludes with section VI.

II. BACKGROUND

A. Blockchain Technology

A blockchain is a decentralized and distributed ledger that securely records transactions and stores information across multiple nodes in a network [13]–[15]. Key aspects of blockchain technology include mining, which is the fundamental mechanism employed by blockchain to secure the network through consensus algorithms [16]. functionality of blockchain provides a foundation for trust, transparency, and security in digital transactions. Immutability is a critical characteristic of blockchain that ensures data integrity and tamper resistance. This property is achieved through two essential components: the storage of the Merkle root and the hash of the previous block. The Merkle tree, a significant data structure in blockchain technology, is vital in maintaining data integrity. It enables unique verification of data blocks without revealing other information [17]. Additionally, the linking mechanism of blockchain, as described by Liao et al. [7], involves creating a chain of linked blocks, as seen in Figure 1, where each block contains a cryptographic

hash of the previous block [18]. This interdependence of blocks ensures that any modification made to a previous block would alter its hash, thereby invalidating all subsequent blocks. This notion of immutability, provided by the storage of the Merkle root and the hash of the previous block, enhances the integrity of the overall blockchain system.

Blockchain technology encompasses various types, including public and private blockchains. Public blockchains, as exemplified by Bitcoin and Ethereum, are open to anyone. Conversely, private blockchains restrict access to a specific group of participants and are commonly used in enterprise settings for enhanced privacy and control [19]. Notably, some blockchains may also incorporate smart contracts. A smart contract is a self-executing computer program that operates on the Blockchain, automatically executing predefined actions according to specified conditions [20]. These various aspects collectively contribute to blockchain technology's effectiveness and widespread adoption.
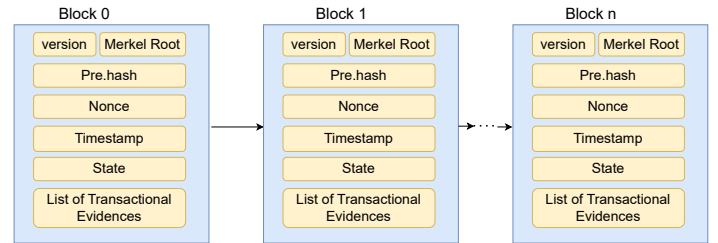

Fig. 1: chain of blocks

B. Provenance

Data provenance refers to understanding where data originates from and its life cycle. It includes metadata describing an end product's origins, history, and evolution. Provenance, also known as lineage, encompasses a wide range of entities, data, processes, activities, and users involved in the entire process. With the exponential growth of digital data being created, copied, transferred, and manipulated through online platforms, provenance has become increasingly important in security. In the context of digital forensics, data provenance plays a critical role in establishing the legitimacy and origin of data, facilitating identification and reuse, and safeguarding the integrity of systems [21], [22].

C. Digital Forensics

Digital forensics deals with retrieving and examining electronic device data. This process comprises five stages: 1) identification, 2) preservation, 3) collection, 4) analysis, and 5) reporting. Potential evidence sources and individuals connected to the investigated device are determined during the identification stage. Preservation involves safeguarding all relevant electronically stored information (ESI) and documenting scene details. In the collection phase, digital information is gathered, creating duplicates for later analysis. The analysis stage involves a thorough search for evidence and its systematic examination.
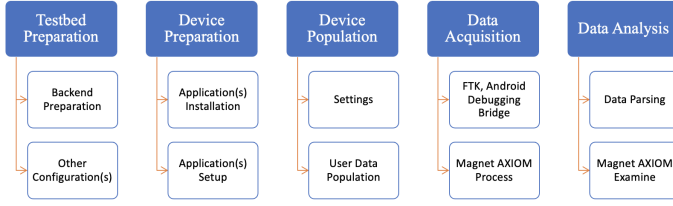
Fig. 2: Conceptual Overview of the Digital Forensics Process

Finally, the reporting stage produces a comprehensive report following NIST guidelines [23].

These five stages provide a systematic methodology for extracting, preserving, and analyzing digital evidence to maintain its integrity and admissibility in legal proceedings. Figure 2 represents a conceptual overview of the digital forensics process used in our work.

III. RELATED WORK

Blockchain technology has been extensively explored for recording data provenance in various domains, including GDPR data collections [24], IoT (elaborated below), supply chain management [25], machine learning [26], cloud computing [27], scientific workflows, legal scenarios and digital forensics [28]–[30]. Systems such as LineageChain [31] and BlockCloud [32] focus on detecting data modification attempts and implementing efficient query techniques and consensus protocols. ProvHL [33] emphasizes access control management and user consent mechanisms. Duong and Dang [34] propose a public-permission model for open-access data. Provenance also holds significant importance in specific fields such as scientific workflows as well. In this context, works like BlockFlow [35], [30] SciLedger [30], SmartProvenance [36], DataProv [37], the work by Nizamuddin *et al.* [38], SciBlock [39], Bloxberg [40], and SciChain [41] introduce specialized approaches incorporating event listeners, voting systems, decentralized databases, timestamp-based invalidation, and unique provenance models.

In recent years, the Internet of Things (IoT) has witnessed a remarkable expansion across various domains [42]. One noteworthy development in the IoT domain is incorporating blockchain-based provenance mechanisms. In the domain of IoT, blockchain-based provenance mechanisms ensure integrity and verifiability through transaction records in the blockchain network [43]. Caro *et al.* [44] developed AgriBlockIoT, a decentralized traceability system for Agri-Food supply chains, utilizing blockchain to record the entire supply chain and provide consumers with product history. Pahl *et al.* [45] integrated IoT edge orchestrations with blockchain-based provenance to address trust concerns by recording origin and actions in the blockchain network. Javaid *et al.* [46] presented BlockPro, a secure IoT framework utilizing blockchain and physically unclonable functions for data provenance and integrity. Ali *et al.* [47] proposed a secure provenance framework for cloud-centric IoT, incorporating blockchain to identify data origin and provide periodic traffic profiles.

Provenance records are essential in digital forensics for preserving evidence integrity and authenticity. They establish data origin, prevent tampering, and ensure a reliable chain of custody. Authors of [1] introduced the IoT forensic chain (IoTFC), a forensic framework powered by blockchain technology, specifically designed to address the challenges of digital forensics in the Internet of Things (IoT) environment. These challenges include ensuring the trustworthiness of evidence items in digital forensics, maintaining continuous integrity checks for evidence items and examination events, providing hash validation for all evidence pieces. The framework emphasizes a comprehensive data provenance architecture and ensures the integrity of examination operations. However, this framework has several limitations, including: it ignores access control, which is a crucial component in managing the integrity and confidentiality of digital evidence. There is a lack of detailed communication between the various framework components, leading to some ambiguities, and the framework lacks evaluation for applicability and data extraction effectiveness.

Borse *et al.* in [3] introduces a novel approach: a hybrid blockchain solution that integrates features from both public and private blockchains. This solution is designed to effectively manage the Chain of Custody (CoC) in order to enhance transparency in evidence handling and transactions. While this method ensures the integrity and security of digital evidence in the field of digital forensics, it primarily focuses on maintaining the CoC within a specific investigative process. However, this approach does not consider access control or offer a comprehensive solution for data provenance in digital forensics.

Ahmed *et al.* [10] proposes a Hyperledger based on a low-cost private blockchain and IPFS for media file tracking as evidence. The implemented access control allows the owner to access all capabilities for handling criminal records while other users have restricted access. However, the design lacks comprehensiveness, focusing on specific scenarios and a data forensic model. The access control model is simplistic, granting the data owner full access across various data forensic stages. Thus, there is a need for a more comprehensive solution that addresses all forensic authorization needs in data provenance and considers the stages of data forensics in the authorization component.

Lone *et al.* [8] and Tsai *et al.* [9] employed Ethereum, a public blockchain, to augment criminal investigations and establish a chain of custody mechanism. However, the utilization of public blockchains raises concerns regarding open-access and potential data privacy breaches

Overall, the utilization of blockchains for provenance in digital forensics often revolves exploiting the immutability feature without considering the unique design requirements specific to the domain. Our work in this paper is distinct in offering a private blockchain-based framework that is customized explicitly for digital forensics.

IV. FORENSIBLOCK FRAMEWORK

A. Motivation and Objective

ForensiBlock addresses the following requirements for digital forensics:

1) Enhanced Evidence Traceability: ForensiBlock ensures enhanced evidence traceability by meticulously logging all steps in creating and modifying files. Additionally, individual tokens are generated for each piece of evidence, enabling seamless linking of items to their sources.
2) Access Control Method: Our access control method caters to investigation stages and considers roles, granting appropriate access levels and enhancing data privacy.
3) Provenance Records: ForensiBlock maintains detailed provenance records or system interactions, promoting transparency and accountability during the investigation.
4) Immutability and Auditability: Leveraging blockchain technology, ForensiBlock ensures immutable and auditable evidence, meeting chain of custody requirements.
5) Fast Extraction with Verification: ForensiBlock facilitates the swift retrieval of case-related information by securely storing it in the system's storage. Furthermore, it implements a robust verification method, ensuring the integrity, reliability, and accuracy of the accessed data.
6) Version Tracking of Data: ForensiBlock simplifies data version tracking and changes for investigators by linking versions together and maintaining a record, guaranteeing accuracy and reliability.

In summary, ForensiBlock incorporates the entire investigation process, from case creation to data extraction, while ensuring secure access.

B. ForensiBlock Architecture

As illustrated in Figure 3, ForensiBlock consists of three main components: blockchain, users, and storage.

- Blockchain: Blockchain is used as the underlying technology to provide a decentralized and tamper-evident ledger for recording all transactions and data changes.
- User Node: The users of the data provenance platform in digital forensics are typically employees or authorized individuals associated with organizations or entities involved in digital forensic investigations.
- Off-chain Storage: The storage component is crucial in securely storing the digital forensic data associated with each case and maintaining the provenance recorded hierarchy for every case file. It receives encrypted data from users and ensures that authorized users can access the data based on the access rights determined by the access control smart contracts. Furthermore, the storage component maintains the provenance records for each case. This allows for faster records extraction and provides detailed data history, ensuring the stored information's integrity and traceability.

C. Blockchain Structure

ForensiBlock utilizes a customized private blockchain with four types of smart contracts that implement specific functionalities and streamline data management.

Tokenized Smart Contract – This smart contract is responsible for creating and managing cases within the blockchain. It also creates tokens for each file associated with a case.

Case Smart Contract– This smart contract is created per each case and contains metadata such as case number, timestamp, the block number of the initial case, current stage, token list, and roles involved. The case-specific smart contract ensures isolation and modularity by containing the specific details of each case. This design enables the streamlined and organized administration of numerous cases, eliminating the necessity for a singular, all-encompassing contract to manage every case.

Access Control Smart Contract– It handles the system's access control mechanisms. This smart contract maintains a list of all cases and determines users' permissions and access rights based on their roles and the case stage. The access control smart contracts interact with the tokenized smart contracts to adjust access levels and validate user access requests.

Provenance Smart Contract– This smart contract facilitates the secure extraction of all related records associated with a case from the storage. Its primary objective is to ensure the integrity of the retrieved data by verifying that the case's records have not been maliciously tampered. The smart contract employs robust verification mechanisms to protect against potential data corruption originating from unreliable or malicious storage sources. The contract ensures that the extracted records remain trustworthy and free from unauthorized modifications by conducting thorough verification checks.
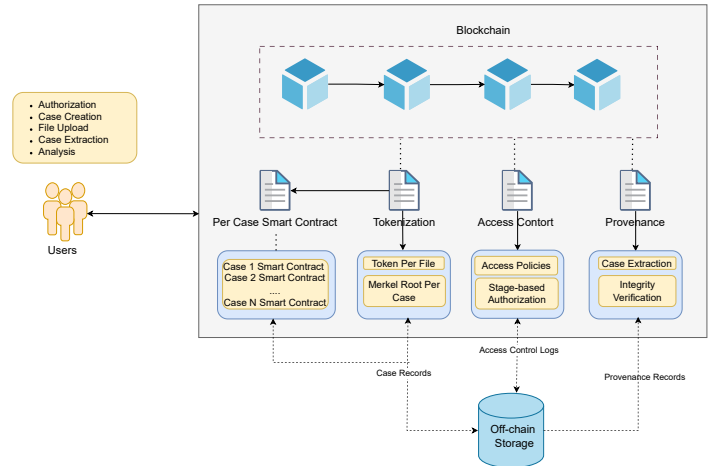

Fig. 3: ForensiBlock Architecture

D. System Phases

The data provenance platform operates through several phases that facilitate user interaction, data storage, retrieval, and analysis. These phases are as follows:

- Phase 1, user registration: Similar to the registration process described in [48], the first phase of our system involves introducing users to the blockchain. This registration process is essential for establishing the user's identity within the system. To achieve this, the system adds the users' public key and performs a setup transaction to complete the registration process.
- Phase 2, uploading data and stage specification: As depicted in Figure 4, a case is created when users send a $T_{InitialUpload}$ transaction. This transaction contains the details of the case, including the case number. Once the blockchain receives this transaction, the tokenized smart contract is triggered, creating a new case smart contract specifically for the case. Additionally, it communicates the case number and current stage to the access control smart contract through $T_{AccessSC}$. Upon receiving $T_{AccessSC}$, the access control smart contract adds the case number to the case list and sets permissions based on the case stage and the roles involved. The tokenized smart contract's output, i.e., case smart contract, along with the $T_{AccessSC}$, is logged on the blockchain. Simultaneously, the tokenized smart sends the encrypted transaction information and the output to the storage for provenance record keeping, while transferring the *EncPkU(time, hashed data, case)* to the user.
- Phase 3, uploading files: Once a case is created, the user can upload related files by sending a $T_{FileUpload}$ per each file. This triggers the tokenized smart contract, which generates a token for the file. The token is then added to the list associated with the case number's smart contract. After logging on the blockchain, the tokenized smart contract forwards the token and the $T_{FileUpload}$ transaction to the storage for provenance and sends the *EncPkU (time, token, case)* to the user.
- Phase 4, data access inquiry: Upon receiving T_{AccReq} from a user, the access smart contract validates the request and assigns appropriate access levels. The T_{AccReq} and the user's access levels are encrypted and transmitted to the storage.
- Phase 5, uploading new edits of previously accessed data: As illustrated in Figure 4, after analyzing the data, the user can send an $T_{Analysis}$ to the blockchain. This action triggers the tokenized smart contract to generate a token for the new file, add it to the list of tokens within the smart contract, and update the relevant case details. The subsequent steps in this phase are similar to phase 2.

Note that, at any phase, a user with specified access privilege can submit a T_{Stage} for changing the case stage. Upon receiving this transaction, the tokenized and access control smart contracts will change the case stage and adjust the rules accordingly. The system also considers the Extraction of provenance records which a user can extract the case records online or offline. To extract it online, the user must submit a $T_{Provenance}$ transaction. Upon receiving $T_{Provenance}$, the provenance smart contract will be provoked and returns all the provenance information related to that specific case.
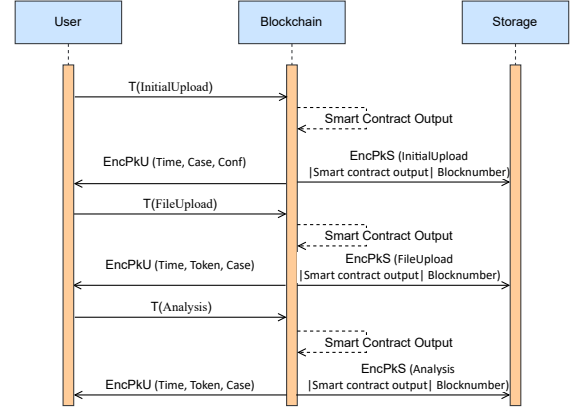

Fig. 4: Transaction flow

E. Protocols

ForensiBlock introduces a set of protocols to enhance provenance records. It achieves this by recording every action, ensuring secure data access tailored for digital forensics, enabling efficient information extraction from dedicated storage, and implementing a robust integrity verification method. Detailed explanations of protocols used in Frensiblock are as follows.

1) Role-Based Access Control with Staged Authorization (RBAC-SA): In digital forensics, available access control methods such as Role-Based [49], Attribute-Based [50], or Rule-Based [51] approaches often have limitations due to unique access control requirements. Specifically, the need for permission changes at different investigation stages can lead to complications when employing Role-Based access control, including potential errors and ongoing challenges [48]. Attribute-Based access control can become complicated due to managing multiple attributes and policies related to evidence. Rule-Based systems may not be adaptable to the dynamic nature of investigation stages, leading to difficulties in updating rules.

To address these limitations, we propose a new protocol called Role-Based Access Control with Staged Authorization (RBAC-SA). RBAC-SA combines role-based and rule-based access control, introducing stages for different investigation phases, i.e., Affidavit Warrant, Investigation, Analysis, Presented in Court, Judgement Day, Case Closed, and Potential Appeal. This approach enables different role permissions at each stage, adding a dynamic aspect to the access control framework.

Our access control smart contract follows a rigorous authentication process, granting user access based on their public key and determining their resource access level. Investigation participants, such as Digital Forensics Examiner, Investigator, Legal Counsel, and Law Enforcement personnel, are assigned specific roles with predefined responsibilities and permissions, ensuring appropriate access privileges based on roles and enhancing security during the investigation. As the investigation progresses, RBAC-SA dynamically adjusts permissions based on predefined rules. For example, only law enforcement and

digital forensics examiners can access the evidence during Affidavit, expanding to other roles in later stages. This fine-grained control over data access by the progression of the investigation enhances security, prevents unauthorized access to sensitive information, and guarantees that only authorized individuals with a legitimate need can access and modify data during each investigation stage. Protocol 1 exemplifies the RBAC-SA mechanism, retrieving access information based on transaction details.

Protocol 1 RBAC-SA Protocol

1: **procedure** RETRIEVEACCESSINFO(transaction)
2:     Read role and stage information from files
3:     Stored_stage = *transaction['current_stage']*
4:     Sender_publik_key = *transaction['sender_public_key']*
5:     **if** *current_stage* is invalid or $\neq$ Stored_stage **then**
6:         **return** 'Invalid stage'
7:     Retrieve *public_key_role* based on Sender_public_key
8:     **if** *public_key_role is None* **then**
9:         **return** 'Access Denied'
10:     Retrieve *public_key_rights* based on *current_stage* and
11:     *public_key_role*
12:     **if** *public_key_rights is None* **then**
13:         **return** 'No access rights'
14:     **return** *public_key_rights*

---

2) Provenance Records Management Protocol: In designing ForensiBlock, our primary goal is to achieve effective storage and extraction of provenance records related to investigations. While blockchain storage suffices for some scenarios, efficient information is crucial for enabling effective analysis, facilitating seamless collaboration, and ensuring proper audit and accountability. Forensiblock incorporates various provenance records, as depicted in Table I. We have devised the Provenance Record Management Protocol to ensure the preservation of these records, track potential malicious modifications, and verify their authenticity. This protocol involves different strategies utilizing blockchain and off-chain storage, which will be further elaborated below.

a) File Version Tokens: The relationship among our data exhibits complexity, as illustrated in Figure 5. The analyzed data can either be independent or dependent on one another. For instance, files A and B were combined to create file AB, while files B and C were utilized to generate file BC2. Subsequently, files A and BC2 were employed to produce ABC2.

To manage this data complexity, we propose a token generation algorithm that generates distinct tokens per file, acting as unique identifiers for file versions. This algorithm creates tokens for two types of files: original files and derived files. When an original file i is added or modified, the token K_i representing the file's version is generated. On the other hand, when a derived file is created based on n initial files with tokens $K_1, K_2, \ldots, K_n$, a derived token, K_{derived}, is generated. This token is created by hashing the tokens of the initial file, i.e.,

$$K_{\text{derived}} = \text{Hash}(K_1||k_2||\ldots||K_n, \text{Time})$$

TABLE I: Provenance Records for the ForensiBlock System

Record	Description
Case Number	Unique identifier for a case
Timestamp	Time of the action/event
Initial Block Number	Block where the initial case transaction was recorded
Current Stage	Current stage of the case
Token List	List of tokens associated with the case
Access Request	User request to access specific case data
Client Info	Information about the user/client
Token dependency	analyzed Tokens derived from a files
Access Validity	Matching system permissions for the access request
Stage Change	Change in the tokenized smart contract's stage
Type of Data Upload	Indicates raw data or analyzed data

Here, $||$ indicates concatenation, and *Time* represents the timestamp of the corresponding transaction. Note that including the time in the hash calculation ensures that even if the same files are combined multiple times, distinct derived tokens will be generated due to the variation in timestamps. This token creation method ensures that the token of the derived file is based on the tokens of its parent files. By applying this algorithm, each case captures the versioning steps and relationships of the files associated with that case.
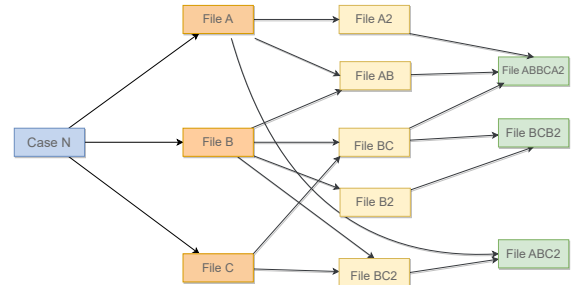

Fig. 5: File dependency example per case

b) Enhancing Provenance Queries: To enhance the extraction of provenance records, two methods are utilized. The first method involves storing the provenance records on the blockchain. The blockchain maintains additional information, such as the unique case number associated with each transaction. This facilitates the extraction of relevant transactions linked explicitly to the desired case. The design incorporates tokens as unique identifiers for each file within a case, which are stored in every case's smart contract. The transactions recorded in the blockchain include these token identifiers, establishing a seamless link between the data and the tokens. This mechanism simplifies the traversal of the blockchain. It enables the con-
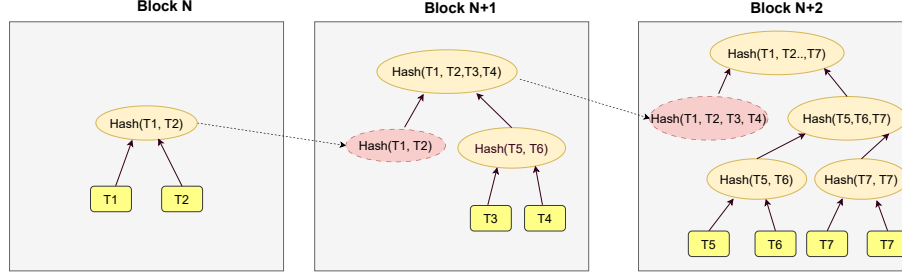
Fig. 6: Merkle Root for Distributed Case Tracking

struction of a hierarchical representation, resembling a tree-like structure, that effectively captures the sequence of edits made to the data over time. The immutable and trustworthy nature of the blockchain ensures the reliability of the extracted provenance records.

The second method includes storing the records in an off-chain storage system, allowing faster information retrieval per case. However, when extracting records from the storage, it is essential to verify their trustworthiness. To address this, we propose that distributed Merkle root for case tracking. Each case added in the block transactions has its own Merkle root, denoted as M_{case}. As shown in Figure 6, this Merkle root is calculated as the hash value of the previous Merkle root associated with the case number, hashed with the transactions added in the current block.

Let M_{prev_case} denote the previous Merkle root associated with the case number, and let T_{block} represent the set of transactions added in the current block. The Merkle root for each case, denoted as M_{case}, is calculated as the hash value of the concatenation of M_{prev_case} and T_{block}:

$$M_{case} = \text{Hash}(M_{prev\_case} \parallel T_{block})$$

This formula ensures that each case added in the block transactions has its unique Merkle root. By hashing the previous Merkle root with the transactions in the current block, we create a tamper-evident structure that links the provenance records of individual cases from one block to the next. Using these two methods, storing provenance records on the blockchain and employing distributed Merkle roots, enhances the system's extraction and verification of provenance records.

c) Verification of Provenance records: The verification process outlined in Algorithm 1 is employed to ensure the integrity of storage provenance records per case. This algorithm constructs a Merkle tree, depicted in Figure 7, and compares the resulting Merkle root with the stored Merkle root. By performing this comparison, it is determined whether any modifications have occurred in the data.

V. EVALUATION

We developed a prototype of the ForensiBlock system using Python to assess the functionality and performance of the proposed algorithm and protocols [1]. The prototype was tested

---

Algorithm 1 Enhanced Storage Verification

procedure STORAGEVERIFICATION(*case_number, sorted_case_list*)
    $M_{case} \leftarrow$ null
    for *block* **in** *sorted_case_record* **do**
        $M_{block} \leftarrow$ null
        $M_{block} \leftarrow$ BlockMerkleRoot(*block_case_record*)
        $M_{case} \leftarrow M_{case} \parallel M_{block}$
    stored_merkle_root $\leftarrow$ Retrieve stored Merkle root for the investigation
    if M_{case} matches stored_merkle_root **then**
        Data integrity verified
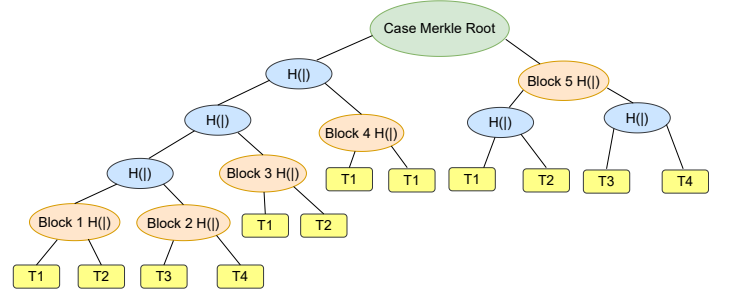   **else**
        Data integrity compromised

Fig. 7: Merkle Tree per each case

on a server running the Ubuntu 18:04 TLS operating system, equipped with an Intel(R) Xeon(R) Gold 6140 2.30GHz CPU and 64 GB of RAM. Notably, running the code to replicate the results presented, necessitates only 4GB of memory. In this section, we conduct an evaluation of the ForensiBlock system, focusing on key aspects such as provenance extraction, smart contract execution, and transaction processing time.

A. Data Generation

To evaluate the ForensiBlock system, we conducted experiments that involved manipulating the number of blocks and cases. We generated a comprehensive list of transactions to simulate the activities within the system. During the transaction generation process, we ensured that each case had an initial transaction known as the $T_{InitialUpload}$. This transaction represented the start of the case and included a randomly

selected stage. In addition to the $T_{InitialUpload}$, we assigned other transactions to cases by randomly selecting transaction types. Once the transactions were generated, we sent them to the network for processing. The transactions were then organized and packed into blocks based on the specified number of transactions per block. By varying the number of blocks and cases in our experiments, we were able to assess the system's performance and behavior under different scenarios.

B. Provenance Extraction

The objective of this evaluation is to assess the extraction time of provenance records associated with a specific case number within the blockchain. Two different methods are explored for achieving this goal. The first method involves a brute-force search of the blockchain. It iterates through the blocks until the desired provenance records are found. However, this approach can be computationally intensive and time-consuming. Alternatively, we can extract the records from the off-chain storage, where the uploaded provenance records are organized based on case numbers. This organized format allows for efficient retrieval of all records associated with a given case number. During the extraction process, the integrity of the retrieved records is validated by verifying the Merkle root of the corresponding case number. This ensures that the provenance information remains intact and untampered.

To evaluate the system, we conducted experiments in which we varied the number of blocks from 0 to 10,000, with 100, 500 and 1,000 randomly distributed cases spread across them. The average time to extract case provenance records from the blockchain is presented in Figure 8. The results demonstrate that the proposed method of extracting records from storage and verifying them is unaffected by the size of the blockchain. However, the brute-force method exhibits a significant increase in extraction time as the blockchain grows. Additionally, as the number of cases increases, the retrieval time of records also increases.

The proposed method efficiently verifies the validity of storage records. However, in cases where the records are invalid, verification with the blockchain itself is necessary. In such scenarios, employing a smart brute-force technique can be beneficial. This technique starts the search from the initial block where a case was created, resulting in faster verification time. Moreover, the smart brute-force technique exhibits less increase in retrieval time compared to other methods when the number of cases increases.

C. Distributed Merkle Root Creation

We examine the effect of the added features for validating provenance records on smart contract processing and creation time. Figure 9 compares the processing time between the provenance system with and without the distributed Merkle root creation. The box plot illustrates the distribution of data using the lower quartile (Q1), median (m or Q2), upper quartile (Q3), and interquartile range
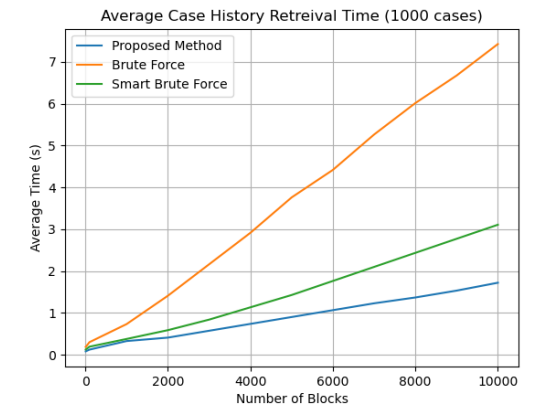
$$IQR = Q3 - Q1 \qquad (1)$$
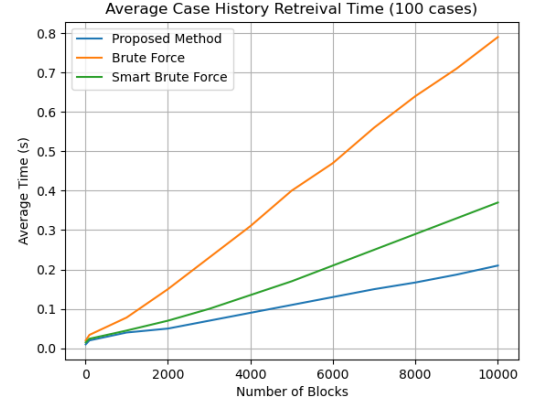
(a) Distribution of 100 cases
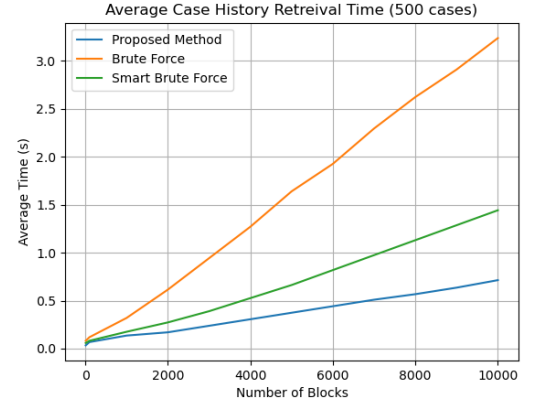
(b) Distribution of 500 cases

(c) Distribution of 1000 cases

Fig. 8: Average time for retrieving a case provenance records

which represents the central 50% of the data. The whiskers extend up to 1.5 times the IQR beyond the box. Any data points outside the whiskers are considered outliers and are plotted individually.

Based on the results shown, the range of values for both cases is almost identical. Although our proposed model has a slightly higher average and maximum value (excluding outliers), the difference is negligible, considering the improved history access time.
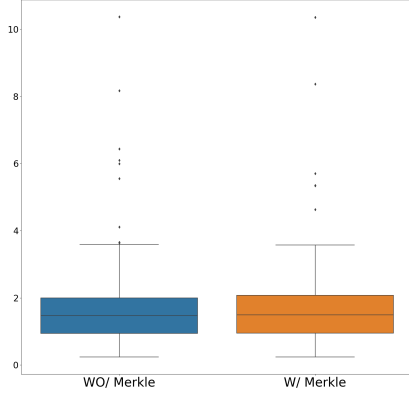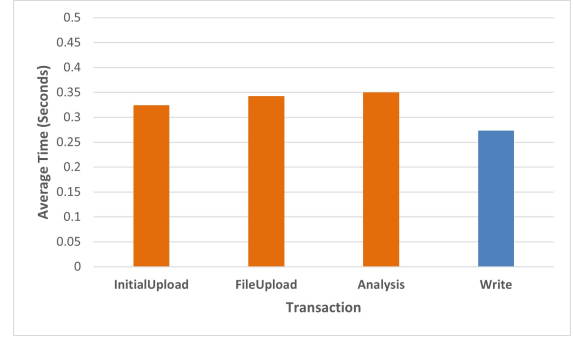

Fig. 9: Average smart contract processing time

(a) Modifications

(b) Retrieval

Fig. 10: Transaction Processing Time

D. Transaction processing time

The transaction processing time in ForensiBlock highlights the distinction between retrieval and modification operations compared to the standard read and write operations used solely for logging purposes. Figure 10a demonstrates the minor variation between these transaction types, which can be attributed to the specific features they introduce. For instance, the transactions related to InitialUpload, FileUpload, and Analysis exhibit higher processing times compared to Write transactions due to their involvement in creating new elements or performing more complex operations. On the other hand, the AccReq transaction in 10b surpasses the read operation in terms of processing time due to its role in accessing and adding user rights.

VI. CONCLUSION

In this paper, we address the importance of provenance in digital forensics and the existing research gap in utilizing blockchain for this purpose. We introduce ForensiBlock, a provenance-driven blockchain solution specifically designed for digital forensics, to overcome the limitations of traditional methods. ForensiBlock ensures comprehensive and transparent record-keeping throughout the investigation process, including steps, extraction, access control, and data version tracking.

The proposed ForensiBlock system incorporates extraction capabilities for timely retrieval of records and a novel access control method to automate digital forensics investigations while ensuring privacy and security. Additionally, we introduce a distributed Merkle tree for verifying the integrity of extracted cases, providing further assurance in the reliability of evidence.

Through the implementation of ForensiBlock and conducting experiments, we demonstrate its capabilities and evaluate its performance. The results highlight the potential of Forensi-Block in enhancing evidence traceability, access control, provenance records, immutability, auditability, fast extraction with verification, and version tracking of data in digital forensics

VII. ACKNOWLEDGMENTS

REFERENCES

[1] S. Li, T. Qin, and G. Min, "Blockchain-based digital forensics investigation framework in the internet of things and social systems," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1433–1441, 2019.

[2] M. S. Akhtar and T. Feng, "Using blockchain to ensure the integrity of digital forensic evidence in an iot environment," *EAI Endorsed Transactions on Creative Technologies*, vol. 9, no. 31, pp. e2–e2, 2022.

[3] Y. Borse, D. Patole, G. Chawhan, G. Kukreja, H. Parekh, and R. Jain, "Advantages of blockchain in digital forensic evidence management," in *Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021)*, 2021.

[4] M. Chopade, S. Khan, U. Shaikh, and R. Pawar, "Digital forensics: Maintaining chain of custody using blockchain," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2019, pp. 744–747.

[5] M. Pourvahab and G. Ekbatanifard, "Digital forensics architecture for evidence collection and provenance preservation in iaas cloud environment using sdn and blockchain technology," *IEEE Access*, vol. 7, pp. 153 349–153 364, 2019.

[6] T. K. Dasaklis, F. Casino, and C. Patsakis, "Sok: Blockchain solutions for forensics," in *Technology Development for Security Practitioners*. Springer, 2021, pp. 21–40.

[7] Z. Liao, X. Pang, J. Zhang, B. Xiong, and J. Wang, "Blockchain on security and forensics management in edge computing for iot: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1159–1175, 2021.

[8] A. H. Lone and R. N. Mir, "Forensic-chain: Ethereum blockchain based digital forensics chain of custody," *Sci. Pract. Cyber Secur. J*, vol. 1, pp. 21–27, 2018.

[9] F.-C. Tsai, "The application of blockchain of custody in criminal investigation process," *Procedia Computer Science*, vol. 192, pp. 2779–2788, 2021.

[10] M. Ahmed, A. R. Pranta, M. F. A. Koly, F. Taher, and M. A. Khan, "Using ipfs and hyperledger on private blockchain to secure the criminal record system," *European Journal of Information Technologies and Computer Science*, vol. 3, no. 1, pp. 1–6, 2023.

[11] M. A. Tasnim, A. A. Omar, M. S. Rahman, and M. Z. A. Bhuiyan, "Crab: Blockchain based criminal record management system," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage: 11th International Conference and Satellite Workshops, SpaCCS 2018, Melbourne, NSW, Australia, December 11-13, 2018, Proceedings 11*. Springer, 2018, pp. 294–303.

[12] S. Bonomi, M. Casini, and C. Ciccotelli, "B-coc: A blockchain-based chain of custody for evidences management in digital forensics," *arXiv preprint arXiv:1807.10359*, 2018.

[13] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–31, 2023.

[14] A. J. Akbarfam, S. Barazandeh, H. Maleki, and D. Gupta, "Dlacb: Deep learning based access control using blockchain," *arXiv preprint arXiv:2303.14758*, 2023.

[15] R. Adhikari and C. Busch, "Lockless blockchain sharding with multiversion control," in *International Colloquium on Structural Information and Communication Complexity*. Springer, 2023, pp. 112–131.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, p. 21260, 2008.

[17] S. Jing, X. Zheng, and Z. Chen, "Review and investigation of merkle tree's technical principles and related application fields," in *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*. IEEE, 2021, pp. 86–90.

[18] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari, and Y. Cao, "A survey on blockchain technology: evolution, architecture and security," *IEEE Access*, vol. 9, pp. 61 048–61 073, 2021.

[19] M. J. Amiri, D. Agrawal, and A. El Abbadi, "Permissioned blockchains: Properties, techniques and applications," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2813–2820.

[20] N. Vu, A. Ghadge, and M. Bourlakis, "Blockchain adoption in food supply chains: A review and implementation framework," *Production Planning & Control*, vol. 34, no. 6, pp. 506–523, 2023.

[21] O. I. Abiodun, M. Alawida, A. E. Omolara, and A. Alabdulatif, "Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey," *Journal of King Saud University-Computer and Information Sciences*, 2022.

[22] B. Pan, N. Stakhanova, and S. Ray, "Data provenance in security and privacy," *ACM Computing Surveys*, 2023.

[23] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Sp 800-86. guide to integrating forensic techniques into incident response," 2006.

[24] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability and provenance tracking," in *Proceedings of the 12th international conference on availability, reliability and security*, 2017, pp. 1–10.

[25] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A blockchain-based framework for supply chain provenance," *IEEE Access*, vol. 7, pp. 157 113–157 125, 2019.

[26] P. Lüthi, T. Gagnaux, and M. Gygli, "Distributed ledger for provenance tracking of artificial intelligence assets," *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers 14*, pp. 411–426, 2020.

[27] Y. Zhang, S. Wu, B. Jin, and J. Du, "A blockchain-based process provenance for cloud forensics," in *2017 3rd IEEE international conference on computer and communications (ICCC)*. IEEE, 2017, pp. 2470–2473.

[28] M. S. Siddiqui, "Activity provenance for rights and usage control in collaborative mr using blockchain," in *2023 20th Learning and Technology Conference (L&T)*. IEEE, 2023, pp. 26–31.

[29] D. Troyer, J. Henry, H. Maleki, G. Dorai, B. Sumner, G. Agrawal, and J. Ingram, "Privacy-preserving framework to facilitate shared data access for wearable devices," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 2583–2592.

[30] R. Hoopes, H. Hardy, M. Long, and G. G. Dagher, "Sciledger: A blockchain-based scientific workflow provenance and data sharing platform," in *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2022, pp. 125–134.

[31] P. Ruan, T. T. A. Dinh, Q. Lin, M. Zhang, G. Chen, and B. C. Ooi, "Lineagechain: a fine-grained, secure and efficient data provenance system for blockchains," *The VLDB Journal*, vol. 30, pp. 3–24, 2021.

[32] 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, 2019.

[33] A. Demichev, A. Kryukov, and N. Prikhodko, "The approach to managing provenance metadata and data access rights in distributed storage using the hyperledger blockchain platform," in *2018 Ivannikov Ispras Open Conference (ISPRAS)*. IEEE, 2018, pp. 131–136.

[34] T. K. Dang and T. A. Duong, "An effective and elastic blockchain-based provenance preserving solution for the open data," *International Journal of Web Information Systems*, vol. 17, no. 5, pp. 480–515, 2021.

[35] R. Coelho, R. Braga, J. M. David, F. Campos, and V. Ströele, "Blockflow: Trust in scientific provenance data," in *Anais do XIII Brazilian e-Science Workshop*. SBC, 2019.

[36] A. Ramachandran and M. Kantarcioglu, "Smartprovenance: a distributed, blockchain based dataprovenance system," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 2018, pp. 35–42.

[37] A. Ramachandran and D. M. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," *arXiv preprint arXiv:1709.10000*, 2017.

[38] N. Nizamuddin, K. Salah, M. A. Azad, J. Arshad, and M. Rehman, "Decentralized document version control using ethereum blockchain and ipfs," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[39] D. Fernando, S. Kulshrestha, J. D. Herath, N. Mahadik, Y. Ma, C. Bai, P. Yang, G. Yan, and S. Lu, "Sciblock: A blockchain-based tamper-proof non-repudiable storage for scientific workflow provenance," in *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2019, pp. 81–90.

[40] K. Wittek, N. Wittek, J. Lawton, I. Dohndorf, A. Weinert, and A. Ionita, "A blockchain-based approach to provenance and reproducibility in research workflows," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2021, pp. 1–6.

[41] A. Al-Mamun, F. Yan, and D. Zhao, "Scichain: Blockchain-enabled lightweight and efficient data provenance for reproducible scientific computing," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1853–1858.

[42] A. J. Akbarfam and K. K. Motarjemi, "Proposing a new protocol for using device-to-device communications in narrowband iot-based systems," in *2021 11th smart grid conference (SGC)*. IEEE, 2021, pp. 1–5.

[43] R. Hu, Z. Yan, W. Ding, and L. T. Yang, "A survey on data provenance in iot," *World Wide Web*, vol. 23, pp. 1441–1463, 2020.

[44] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in agri-food supply chain management: A practical implementation," in *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*. IEEE, 2018, pp. 1–4.

[45] C. Pahl, N. El Ioini, S. Helmer, and B. Lee, "An architecture pattern for trusted orchestration in iot edge clouds," in *2018 third international conference on fog and mobile edge computing (FMEC)*. IEEE, 2018, pp. 63–70.

[46] U. Javaid, M. N. Aman, and B. Sikdar, "Blockpro: Blockchain based data provenance and integrity for secure iot environments," in *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems*, 2018, pp. 13–18.

[47] S. Ali, G. Wang, M. Z. A. Bhuiyan, and H. Jiang, "Secure data provenance in cloud-centric internet of things via blockchain smart contracts," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*

(SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2018, pp. 991–998.

[48] A. J. Akbarfam, S. Barazandeh, H. Maleki, and D. Gupta, "Dlacb: Deep learning based access control using blockchain."

[49] R. S. Sandhu, "Role-based access control," in *Advances in computers*. Elsevier, 1998, vol. 46, pp. 237–286.

[50] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[51] S. A. Afonin, "Performance evaluation of a rule-based access control framework," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2016, pp. 1414–1418.