information

MDPI

Article

ForensicTransMonitor: A Comprehensive Blockchain Approach to Reinvent Digital Forensics and Evidence Management

Saad Said Alqahtany † and Toqeer Ali Syed *,†

Faculty of Computer and Information System, Islamic University of Madinah, Madinah 42351, Saudi Arabia
* Correspondence: toqeer@iu.edu.sa
† These authors contributed equally to this work.

Abstract: In the domain of computer forensics, ensuring the integrity of operations like preservation, acquisition, analysis, and documentation is critical. Discrepancies in these processes can compromise evidence and lead to potential miscarriages of justice. To address this, we developed a generic methodology integrating each forensic transaction into an immutable blockchain entry, establishing transparency and authenticity from data preservation to final reporting. Our framework was designed to manage a wide range of forensic applications across different domains, including technology-focused areas such as the Internet of Things (IoT) and cloud computing, as well as sector-specific fields like healthcare. Centralizing our approach are smart contracts that seamlessly connect forensic applications to the blockchain via specialized APIs. Every action within the forensic process triggers a verifiable transaction on the blockchain, enabling a comprehensive and tamper-proof case presentation in court. Performance evaluations confirmed that our system operates with minimal overhead, ensuring that the integration bolsters the judicial process without hindering forensic investigations.

Keywords: digital forensics, forensics blockchain, forensics integration with blockchain, forensics analysis with blockchain

Citation: Alqahtani, S.S.; Syed, T.A. ForensicTransMonitor: A Comprehensive Blockchain Approach to Reinvent Digital Forensics and Evidence Management. *Information* **2024**, *15*, 109. https://doi.org/10.3390/info15020109

Academic Editor: Rui Zhang

Received: 2 January 2024
Revised: 3 February 2024
Accepted: 8 February 2024
Published: 13 February 2024

1. Introduction

There has been a steady increase in interest in blockchain technology over the past few years. Many industries, including finance, healthcare, supply chains, and real estate, are exploring its potential applications. The financial sector has been at the forefront of blockchain adoption, with many banks and financial institutions exploring and implementing blockchain solutions for various use cases, including cross-border payments and fraud prevention.

Blockchain technology encompasses a set of distinctive features that redefine how data and transactions are managed. Its decentralization means that control is not concentrated in a single entity or location, enhancing its resilience against censorship and tampering. Transparency is a hallmark feature, as every transaction is openly visible to all participants in the network. This not only promotes trust but also ensures accountability as participants can verify transactions independently. Immutability, a result of robust cryptographic techniques, guarantees that once data are recorded, they cannot be altered or deleted easily. This feature is crucial for maintaining the integrity of the information stored on the blockchain.

Over the past few years, businesses across various sectors have increasingly invested in blockchain technology, recognizing its transformative potential. The market for blockchain technology is increasing exponentially. In 2022, the worldwide market for blockchain technology was estimated at USD 11.14 billion. It is anticipated to expand from USD 17.57 billion in 2023 to USD 469.49 billion by 2030, demonstrating a Compound Annual Growth Rate (CAGR) of 59.9 percent throughout the forecast period. In terms of market

share, North America was the leading region in 2022, holding 47.13 percent of the global market [1]. In the financial services sector, blockchain has already achieved mainstream adoption, as acknowledged by 96 percent of experts. Additionally, the manufacturing sector is set to experience substantial growth in blockchain adoption, with a projected growth rate of 73 percent between 2023 and 2026. The primary driver of blockchain adoption is its ability to establish provenance, and 52 percent of experts believe it will be essential for verifying customers' identities in the future [2].

Digital forensics, sometimes known as "cyber forensics" or "computer forensics", is an interdisciplinary domain encompassing the gathering, safeguarding, scrutiny, and introduction of electronic evidence within the legal framework. Its primary objective is to unveil digital data originating from diverse electronic devices and systems, including computers, mobile phones, servers, and network infrastructure.

Blockchain is very widely used in computer forensics, whether it be the cloud, IoT, or other technology; however, there is no comprehensive framework that could store all forensic transactions on the blockchain. In this paper, we propose a generic framework for reinventing forensic applications that will be integrated with blockchain as a pluggable architecture. This pluggable architecture will accommodate all the applications of forensics, such as data preservation, data acquisition, analysis, and documentation.

Considerable effort has been exerted to secure various transactions of computer forensics with blockchain, such as storing chain-of-custody-related transactions in blockchain and storing IoT- and cloud-related transactions in blockchain. In this regard, the following paragraphs discuss the current contributions and their proposed solutions to various problems in these areas.

1.1. Chain-of-Custody Integration with Blockchain

The existing research suggests that blockchain technology has been widely used in digital forensics in various ways. First is the integration of blockchain technology into cloud forensics, offering methods and architectures that enhance the credibility, effectiveness, and security of forensic investigations in digital environments mostly related to the chain of custody (CoC). The central focus of [3–15] was the utilization of blockchain technology for ensuring the integrity, traceability, and privacy of digital evidence and CoC in digital forensics. Some of the solutions for specific use cases such as image forensics, healthcare, and finance were generally discussed, as well as blockchain integration with computer forensics in the context of CoC. Ali et al. [3] introduced a novel technique that integrates fuzzy hashing within blockchain structures, primarily used for image forensics. Both Yan et al. [4] and Silva et al. [5] presented frameworks based on blockchain technology that emphasize the importance of traceability and integrity in the chain of custody. Lone et al. [6] proposed a forensic chain model in Hyperledger Composer, providing tamper resistance. Al-Khateeb et al. [7] discussed a CoC based on a distributed ledger, focusing on the admissibility of digital evidence in legal scenarios. Gopalan et al. [9] explored the enhancement of the chain-of-custody process using blockchain, emphasizing its importance in the legal context. Li et al. [8] presented the LEChain system, which oversees the complete legal evidence management process, from evidence collection to court trials. However, evidence for their claims was not provided in their paper.

1.2. Blockchain in IoT Digital Forensics

The intersection of Internet of Things (IoT) and blockchain technology, exploring how blockchain can play a crucial role in ensuring the reliability and legal admissibility of digital forensic processes within the IoT domain, is also an important aspect. Studies such as [16–28] dove into the realm of the Internet of Things (IoT) and how blockchain can assist in ensuring the reliability of digital forensic processes within this domain. Brotsis et al. [16] specifically targeted smart home environments, using blockchain to ensure the legal admissibility of evidence. Liao et al. [17] and Kamal et al. [18] provided comprehensive overviews of the intersection of IoT, forensics, and blockchain, highlighting

security challenges and potential future directions. Li et al. [19] presented the IoT Forensic Chain, emphasizing the importance of a transparent audit trail in forensic investigations for IoT. Kumar et al. [20] introduced the Internet-of-Forensics framework designed specifically for IoT forensics, while Ryu et al. [21] focused on the decentralization and integrity advantages offered by blockchain in the IoT domain. Le et al. [22] prioritized identity privacy in their IoT forensics blockchain-based framework.

1.3. Blockchain and Cloud Forensics

Blockchain forensics and cloud forensics are specialized branches of digital forensics that focus on the investigation, analysis, and preservation of digital evidence within the contexts of blockchain technology and cloud computing, respectively. The studies [29–33] primarily dealt with integrating blockchain technology into cloud forensics. These papers presented methods and architectures that leverage blockchain to enhance the credibility, effectiveness, and security of forensic investigations in cloud environments. Whether validating the existence of process records, comparing forensic tools, or logging and preserving admissible evidence, these studies underline the benefits of using blockchain in a cloud ecosystem, especially concerning the chain of custody.

1.4. Specialized Forensic Systems and Applications

Specialized forensic systems and applications are custom-built tools and software designed for precise tasks within forensic science. Refs. [11,33–44,44–50] presented blockchain-based solutions for specialized forensic systems, such as vehicular digital forensics, medical forensics, IoT digital forensics, and multimedia investigations. This research demonstrates how blockchain can safeguard data integrity, maintain anonymity, provide efficient evidence tracking, and ensure the secure exchange of investigation details among various stakeholders. The studies also emphasize the importance of combining blockchain with other technologies, such as fuzzy hashing and Hyperledger Sawtooth, to enhance their systems' efficacy.

While the aforementioned papers focused on solutions tailored to the CoC, IoT, and cloud environment within the realm of digital forensics and the chain of custody, our work delves into a comprehensive implementation strategy specific to cloud, IoT, and specialized forensic applications in digital forensics investigations.

Addressing these challenges, we introduce a groundbreaking methodology that interlinks every forensic transaction, from preservation to documentation. By embedding each transaction as an immutable entry within a blockchain, we ensure the transparency and authenticity of every action taken during an investigation. This approach not only provides a linear and indisputable timeline of events but also guarantees the sanctity of the data at each phase. As shown in Figure 1, our proposed solution is generic for all kinds of forensics applications. Thus, any kind of forensics transaction can be connected with the smart contract API, and the complete process of the forensics investigation can be recorded.

Central to our approach is the deployment of smart contracts that act as the bridge between forensic applications and the blockchain. Through meticulously designed APIs, every forensic action triggers a corresponding transaction on the blockchain. This ensures that the entirety of a forensic case, from data acquisition to the final report, can be presented in court as an unbroken chain of verifiable events, resistant to tampering or disputes.

The performance evaluation of our solution revealed that the addition of smart contracts to forensic applications does not introduce significant overhead. The system operates seamlessly, with minimal latency, ensuring that forensic experts can continue their investigations without any undue burden. In essence, this integration elevates the reliability and verification of computer forensic investigations, offering an indisputable record that strengthens the judicial process.

Paper Organization:

The rest of the paper is organized as follows. Section 2 discusses the forensics and blockchain background. Section 3 provides the proposed generic solution for forensics with blockchain. Section 4 presents the implementation, and, finally, Section 5 explains the performance results of the proposed solution on blockchain.

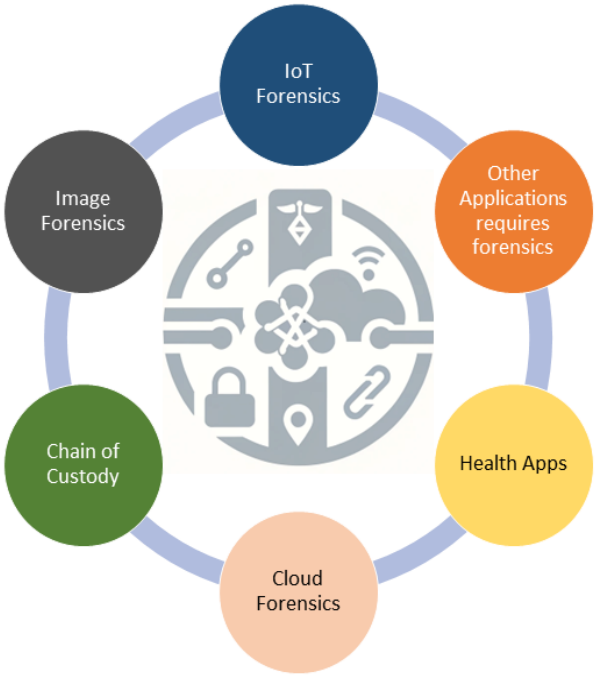

Figure 1. The Proposed generic framework that can encapsulate all types of forensics transactions in blockchain.

2. Background

2.1. Digital Forensics

Digital forensics, also known as computer forensics, is a critical process in the investigation and prevention of cybercrime. It involves the systematic collection, analysis, and preservation of electronic evidence and is particularly essential in cases where digital devices such as computers, smartphones, and storage devices play a role.

The process of digital forensics encompasses several key phases. Initially, it involves the identification of potential evidence sources, which could range from computers and servers to smartphones and other digital devices. Following this, the preservation phase ensures that the identified evidence is kept in its original state, maintaining the data's integrity during the investigation. The collection phase then involves the use of specialized tools and techniques to gather evidence, which might include creating an exact copy of a hard drive or extracting data from a mobile device [51].

Once the evidence is collected, the analysis phase begins. Here, investigators scrutinize the data to uncover evidence pertinent to the crime, searching for specific files, emails, logs, or other relevant data. The findings from this analysis are then compiled and presented, often in a legal setting, in a form that may include reports, presentations, or testimonies. The final closure of the investigation marks the end of the process, where the case is concluded and the collected data are either stored for future reference or securely destroyed [52].

Digital forensics faces several challenges, including the sheer volume of data on digital devices, which can make analysis time-consuming. The presence of encrypted data poses another challenge, requiring decryption keys for analysis. Additionally, the use of cloud storage can complicate matters, as data may be stored outside the investigator's jurisdiction. The field also needs to adapt continually to rapid technological changes.

The evidence collection process in digital forensics starts with identifying potential digital evidence sources, such as computers, servers, mobile devices, storage media, and

network logs. The evidence must then be preserved to prevent any tampering, often involving the creation of a forensic copy of the original data using write-blocking technology. Documentation is crucial at every stage, recording details like the evidence location, involved individuals, collection date and time, and tools and techniques used. After collection, maintaining the integrity of the original evidence is critical, and this could involve creating a complete copy of a hard drive or extracting specific files and logs. Hash values are generated for the collected evidence to verify its integrity later. The chain of custody records the movement of the evidence, documenting every individual who handled it and the dates and times of transfers. Forensic experts then analyze the collected data, with the findings documented in a comprehensive report detailing the methodologies, evidence, and conclusions drawn [53].

The concept of the chain of custody (CoC) is pivotal in digital forensics. It is a legal and procedural process that ensures the chronological documentation and integrity of evidence, whether physical or digital, as it changes hands in legal or investigative contexts. The chain of custody is vital for several reasons. It adds credibility to court-presented evidence by establishing a clear record of the evidence's history. It also ensures the admissibility of evidence in legal proceedings, as improperly tracked and authenticated evidence may be deemed inadmissible. Moreover, it guarantees the evidence's integrity, ensuring it has not been tampered with or altered during collection, storage, and handling. Lastly, a transparent chain of custody process promotes transparency and accountability, reducing the risk of evidence mishandling or misconduct [54].

2.2. Blockchain

Blockchain technology, known for its decentralization, transparency, and security, has found applications in various domains beyond its initial use in cryptocurrency. In healthcare, blockchain provides a secure platform for managing patient records, ensuring data integrity and accessibility while maintaining patient confidentiality [55]. In vehicle tracking, it offers a reliable and tamper-proof system for logging vehicular data, essential for fleet management and regulatory compliance [56]. Blockchain's application in property registration introduces an immutable record-keeping system, significantly reducing fraud and streamlining property ownership transfers [57]. In the realm of augmented reality, blockchain facilitates collaborative experiences by securely managing digital assets and interactions [58]. Lastly, its role in Android malware analysis is groundbreaking; blockchain can be utilized for securely sharing data across multiple entities, enhancing the detection and analysis of malicious software in Android devices, thus bolstering cybersecurity measures [59,60].

Components of Blockchain

Blockchain technology, a cornerstone of modern digital transactions, is characterized by its unique structure and operational mechanisms. It comprises a series of blocks, each containing a collection of transactions or data. These blocks are chronologically linked to form the blockchain, reflecting the system's namesake [61].

Central to blockchain's appeal is its decentralized nature, setting it apart from traditional centralized systems where control is vested in a single entity. Instead, blockchain relies on a network of nodes, or computers, which collectively participate in validating and recording transactions. This decentralized framework not only bolsters security but also enhances the system's resilience against disruptions [62].

At the heart of blockchain functionality are consensus mechanisms, which are crucial for validating transactions and achieving agreement on the blockchain's state process. We used the Raft consensus mechanism for our solution of integrating forensics with blockchain [63].

Another key feature of blockchain is the use of cryptographic hash functions. Each block contains a cryptographic hash of the preceding block, creating an unbreakable link.

These hash functions are essential for maintaining the data's integrity and the sequential order of the blocks [64].

Blockchain technology also employs a distributed ledger system. Every participant in the blockchain network holds a copy of the entire ledger, which is continually updated in real time with new transactions. This widespread distribution of the ledger ensures transparency and significantly reduces the chances of data manipulation or fraud.

Smart contracts are another integral component of blockchain. These are self-executing contracts with the terms of the agreement embedded in code. They facilitate the automation of processes and transactions when predetermined conditions are met, with Ethereum being a notable platform that supports smart contract creation [65].

The blockchain ecosystem includes both public and private blockchains. Public blockchains, like Bitcoin and Ethereum, are open to all, while private blockchains restrict access to authorized participants only. Private blockchains are commonly used in business settings for applications such as supply chain management. In this solution, we utilized a permissioned blockchain network setup [66].

A defining feature of blockchain is its immutability. Once data are entered into a block and the block is added to the blockchain, altering or deleting these data become exceedingly difficult. This characteristic provides a high level of security and trust, making blockchain a reliable system for various applications [67].

Blockchain technology finds diverse applications across several sectors, including finance, supply chain management, healthcare, identity verification, and real estate. It is particularly advantageous in scenarios where multiple parties need to share data and where trust is of paramount importance [68].

In summary, blockchain represents a transformative technology that offers secure, transparent, and decentralized methods for recording and verifying transactions. Its components synergistically create a system that is resistant to tampering and reliable for a multitude of applications.

3. Proposed Solution

As discussed in the introduction and problem statement, computer forensics can be divided into four main parts, which are data preservation, acquisition, analysis, and documentation. In this proposed solution, a complete framework is given for the implementation of computer forensic tools. This architecture is completely pluggable. All the transactions are stored in the blockchain. The transactions related to each part of computer forensics are stored in separate blocks to avoid ambiguity, as shown in Figure 2. A lot of work has been carried out previously on computer forensics, as discussed earlier in the background section. However, no study has provided a complete architecture for computer forensics. The proposed solution given in this paper restructures computer forensics applications, integrating them with blockchain to record each and every transaction of a forensics case. It provides a foolproof architecture that prevents the data from being altered.

3.1. Data Preservation with Blockchain

Data preservation with blockchain means using the blockchain to timestamp the exact moment when evidence is preserved. This can be useful for proving that evidence was collected at a specific time. When evidence is preserved, a cryptographic hash of the forensic image or clone is generated, and this hash, along with a timestamp, is added to the blockchain. This becomes an immutable record of when the evidence was preserved and its state at that time.

The diagram in Figure 3 illustrates the step-by-step interactions between a forensic expert, the original evidence (typically a digital device), and the blockchain network. Initially, the forensic expert creates a digital clone or image of the device, ensuring that the original evidence remains untouched. The device acknowledges this and sends back this cloned image to the expert. Next, for added security, the expert ensures that the device is isolated, typically by disconnecting it from any networks. The device confirms

this isolation status back to the expert. Following this, the forensic expert documents the current state of the device, capturing all the essential details such as active connections and running processes. The device provides these state details to the expert. Once these preliminary forensic steps are completed, the expert then records all this information, along with the chain-of-custody details, on the blockchain network. This integration with the blockchain ensures that the evidence's preservation process is transparent, immutable, and verifiable. The blockchain network, upon successfully recording this transaction, sends confirmation back to the forensic expert. This entire sequence ensures that the forensic evidence and its handling details are securely and transparently stored, leveraging the power and trustworthiness of blockchain technology. The complete procedure of the preservation and its related steps in our proposed model are formally described below.

Figure 2. Complete framework of computer forensics transactions integrated with blockchain.

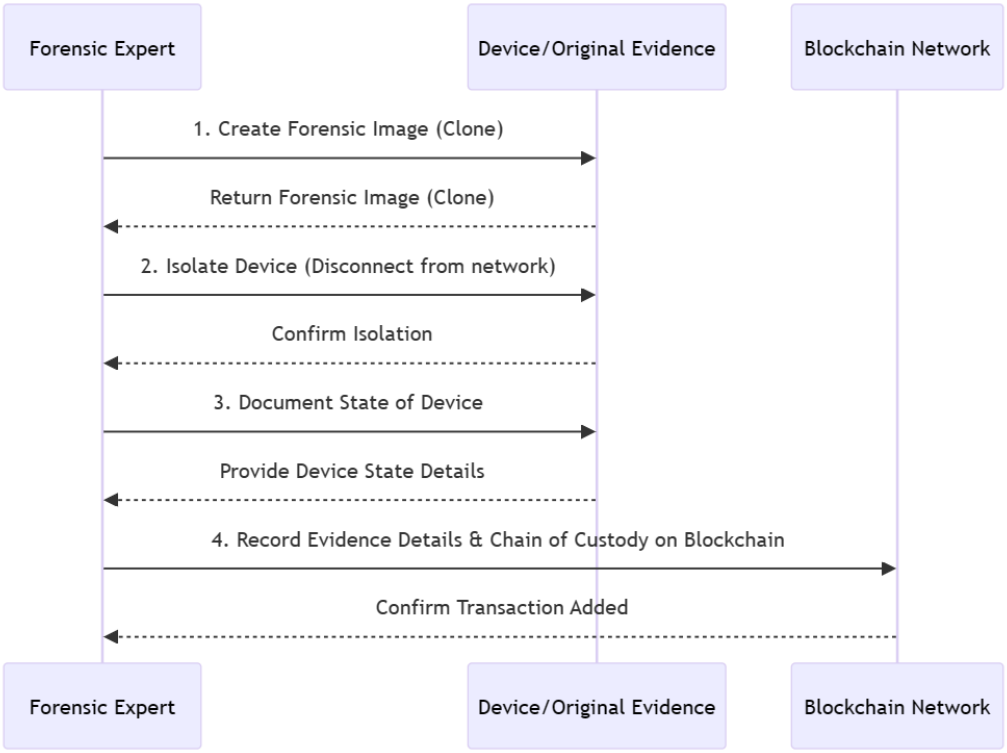

Figure 3. Sequence diagram for data preservation.

Formal Language

The following formal model describes a forensic evidence preservation process integrated with blockchain technology, ensuring the integrity and authenticity of digital evidence. Initially, the model defines creating a forensic image (E') of the original evidence (E) using a function $F(E)$. It then outlines the process for isolating the evidence device $(I(E))$, which returns 1 if successful and 0 otherwise, and documenting the state of the device $(S(E))$ through a function $D(E)$ that captures all relevant metadata and running processes. The chain of custody (C) is maintained by appending the list with each entity (X) that handles the evidence. This process is encapsulated in a blockchain transaction (T) that includes the forensic image, isolation status, documented state, and chain of custody. Each block (B_n) in the blockchain contains this transaction data, the hash of the previous block $(H(B_{n-1}))$, and the transaction hash (H(T)), ensuring the continuity and immutability of the blockchain. The block is then added to the blockchain after consensus, securing the forensic evidence within the blockchain's tamper-evident and transparent ledger. This integration not only preserves the evidence but also enhances its credibility by leveraging blockchain's inherent security features.

By following this model, any alterations to the evidence preservation process would alter $H(T)$, and any attempt to alter past transactions would disrupt the chain, making the attempt evident due to the properties of blockchain. This ensures transparency, integrity, and non-repudiation in the forensic evidence preservation process.

Preservation Transactions Monitoring Framework

Definition.
Let E be the original evidence. Let F(E) be the function that creates a forensic image or clone of E. The output is the cloned evidence, E′, where E′ = F(E).
Forensic Imaging Let E be the original evidence. Let F(E) be the function that creates a forensic image or clone of E. The output is the cloned evidence, E′, where E′ = F(E).
Device Isolation. Let I(E) be the function that isolates the device or data. If E is isolated successfully, I(E) returns 1; otherwise, it returns 0.
Device State Documentation. Let S(E) be the state of the device, which includes all connections, running processes, and other relevant metadata. The documentation function D(E) returns S(E), a complete documented state of E.
Chain of Custody. Let C be the chain of custody, a list of entities that have handled the evidence. Whenever an entity, say, X, handles the evidence, C is updated as C = C + X.
Blockchain Transaction. Let B_n be the n-th block in the blockchain. $B\_n = T, H(B\_n - 1), H(T)$, where T represents the transaction data, in this case consisting of E′, I(E), S(E), and C; H is a cryptographic hash function; $H(B\_n - 1)$ is the hash of the previous block, ensuring the immutability of the blockchain; and $H(T)$ is the hash of the transaction data, ensuring the integrity and authenticity of the evidence and its preservation process.
Integration with Blockchain. First, a new transaction T is created with the following data: Forensic image, E';
Device isolation status, $I(E)$;
Documented state, $S(E)$;
Chain of custody, C.

Then, this transaction is added to a new block B_n. $H(T)$ is calculated and included in the block. The hash of the last block in the blockchain $H(B\_n - 1)$ is retrieved and included in the block. After retrieval, consensus is achieved for the block to become part of the blockchain network.
Finally, B_n is added to the blockchain, thereby recording the forensic evidence preservation transaction.

Figure 4 depicts a flowchart detailing a computer forensics preservation process integrated with blockchain technology for the secure and verifiable storage of forensic data. Each transaction of our generic forensics module is transferred to selected peers via the blockchain network. The endorsement by selected peers section outlines the steps of the orderer nodes in generating blocks; hash calculations; and reaching consensus (using mechanisms like Raft, Kafka, and Solo) for data validation, while interacting with components like the membership service, smart contract peer, and certification authority within the blockchain network. The preservation phase identifies the evidence, creates forensic copies, and ensures secure storage and handling, with a focus on maintaining the chain of custody. All the transactions related to these processes are stored in a CouchDB database, which is part of the blockchain network, to ensure the integrity and verifiability of the forensic data.

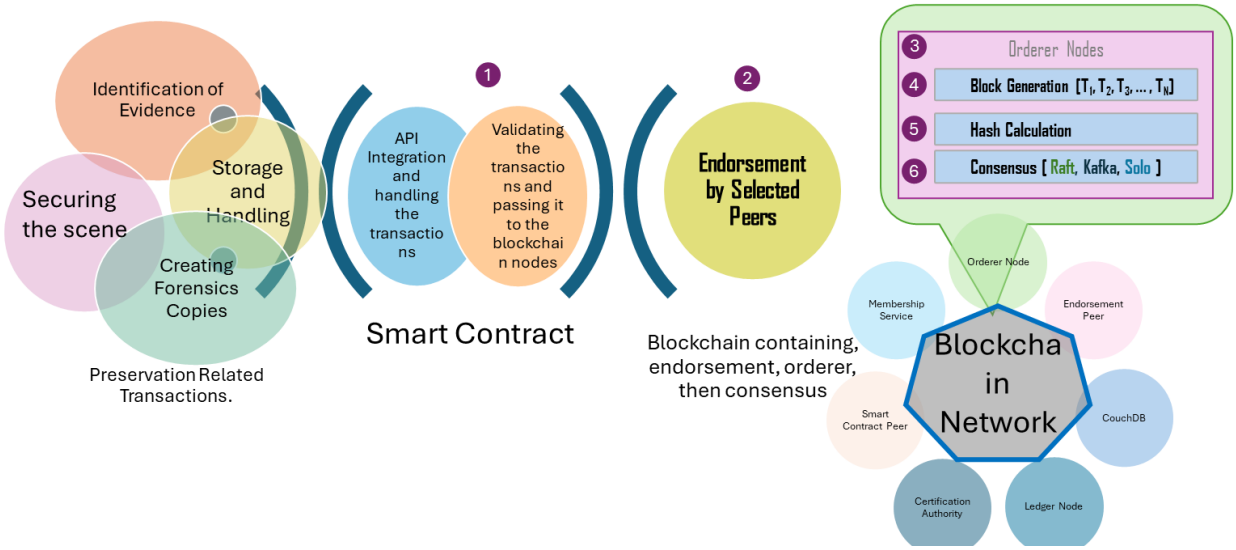

Figure 4. Diagram of integration of data preservation with blockchain.

3.2. Data Acquisition with Blockchain

Integrating data/evidence acquisition means documenting each piece of acquired evidence in the blockchain. This provides a clear record of what was acquired and when. For each piece of acquired evidence, a cryptographic hash is generated. Then, the hash, along with the details of the evidence (e.g., type of device and storage size) and a timestamp, is added to the blockchain.

Figure 5 highlights the legal considerations and the verification of data integrity and continuity in the acquisition phase. Several hashes of the data are taken from time to time in the acquisition stage to ensure the integrity of the data. If any two hashes do not match, it shows that the data have been compromised.

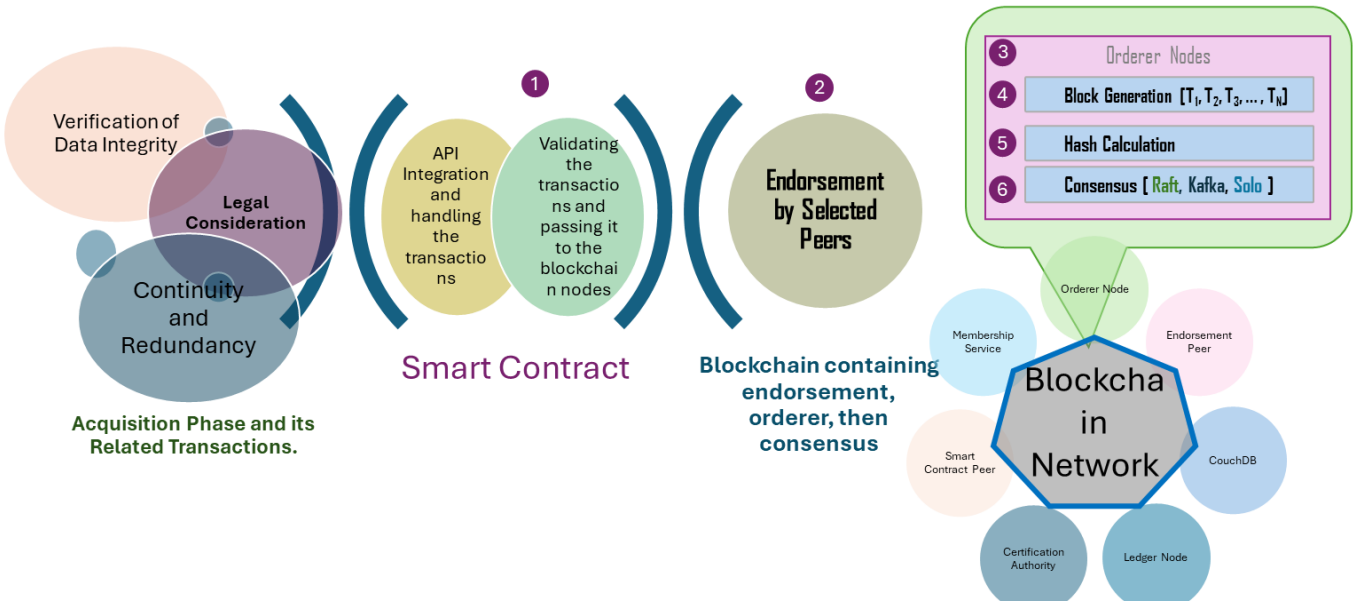

Figure 5. Diagram of integration of data acquisition with blockchain.

For every forensics case, first of all a copy of the original data is created to preserve the data; then, a name is given to the case and is stored on the blockchain along with a unique case ID, indicating the initiation of that case. Several hashes of the data are taken from time to time in the acquisition stage to ensure the integrity of the data. If any two hashes do not match, it shows that the data have been compromised. The data are analyzed with different methods, as explained in the figure. Finally, the documentation of the whole case is carried out, and all the related transactions are recorded in CouchDB.

Formal Definition of Acquisition Transactions

The following formal model outlines a comprehensive forensic acquisition process designed for integrating digital evidence transactions into a blockchain. It begins with acquiring data (D) from various sources (S) using specialized forensic tools (T). This includes making a bit-for-bit copy of data from devices or storage media (Dcopy); capturing live system data like RAM contents (Dlive); handling different storage media types such as HDD, SSD, USB, and CD/DVD (Dm); and extracting data from cloud sources or backup repositories (Dcloud). After acquisition, a transaction record (R) is created, encapsulating metadata about the data (including type, source, size, and timestamp) and a hash of all acquired data (h) using a hash function (H). This hash is a composite of the copied data, live data, data from various media, and cloud data, ensuring the integrity and authenticity of the evidence. The record also includes a reference to the previous transaction on the blockchain (Tprev). The transaction record (R) is then added to the blockchain within a new block (B), updating Tprev to reflect the latest transaction. This process not only secures the forensic data within the immutable and transparent structure of the blockchain but also provides a verifiable chain of custody for the digital evidence, enhancing its reliability and admissibility in legal contexts.

Acquisition Transactions Monitoring Framework

D: Data to be acquired.
S: Source of data (device or storage medium).
T: Specialized tools for copying.
L: Live system data, e.g., RAM contents.
M: Storage media types, e.g., HDD, SSD, USB, CD/DVD.
C: Cloud sources or backup repositories.
B: Block on the blockchain.
H: Hash function.
Tprev: Previous transaction on the blockchain.
Procedure:
Bit-for-bit copy acquisition: Acquire D from S using tool T,
Dcopy=T(S). Live System Data Acquisition:
If S is a live system, acquire L D live=T(L). Handling various storage media:
For every type in M, acquire Dm using T appropriate for type M. Extracting from cloud or backup repositories:
If S is a cloud or backup repository, Dcloud=T(C).
Storing on the blockchain:
Create a transaction record R that contains
metadata about D (type, source, size, timestamp, etc.); a hash of D using H
(h=H(Dcopy + Dlive + Dm + Dcloud)); and a reference to Tprev.
Add R to the blockchain by placing it in B.
Update Tprev to point to the latest transaction.

The sequence diagram in Figure 6 visualizes the process of forensic evidence acquisition integrated with a blockchain system. The process begins with a user initiating the data acquisition through a specialized forensic tool. This tool communicates with various data sources, generalized as 'Storage' in the diagram. First, the tool acquires a bit-for-bit copy of the device or storage medium. Then, if applicable, it extracts data from live systems, capturing possibly volatile contents such as RAM data. Subsequently, the tool acquires data from different storage media types, ensuring compatibility with sources like HDDs, SSDs, or USBs. Finally, it accesses cloud or backup repositories to extract any relevant data. Once all data sources have been interrogated and the required data have been compiled, the tool computes a cryptographic hash of the acquired data to ensure their integrity. The user is then notified that the acquisition and hashing are complete.

3.3. Analysis-Related Transactions with Blockchain

Evidence analysis with blockchain refers to keeping track of all analysis steps and findings in the blockchain. This can demonstrate that proper procedures were followed and can provide an immutable record of the analysis results. As each analysis step is performed, the action, the tool used, and any findings are documented. Also, a cryptographic hash of this documentation is generated. After generating the hash, it is added, along with a description of the analysis step and a timestamp, to the blockchain.

Figure 7 details the use of a smart contract for forensics analysis, including hashing and integrity verification. Tools are used for file and artifact analysis, data correlation and reconstruction, and reporting and documentation, with each action recorded on the blockchain. The analysis column details the use of a smart contract for forensics analysis and includes sub-phases like data decryption, data triage, keyword and pattern searching, and prioritization.

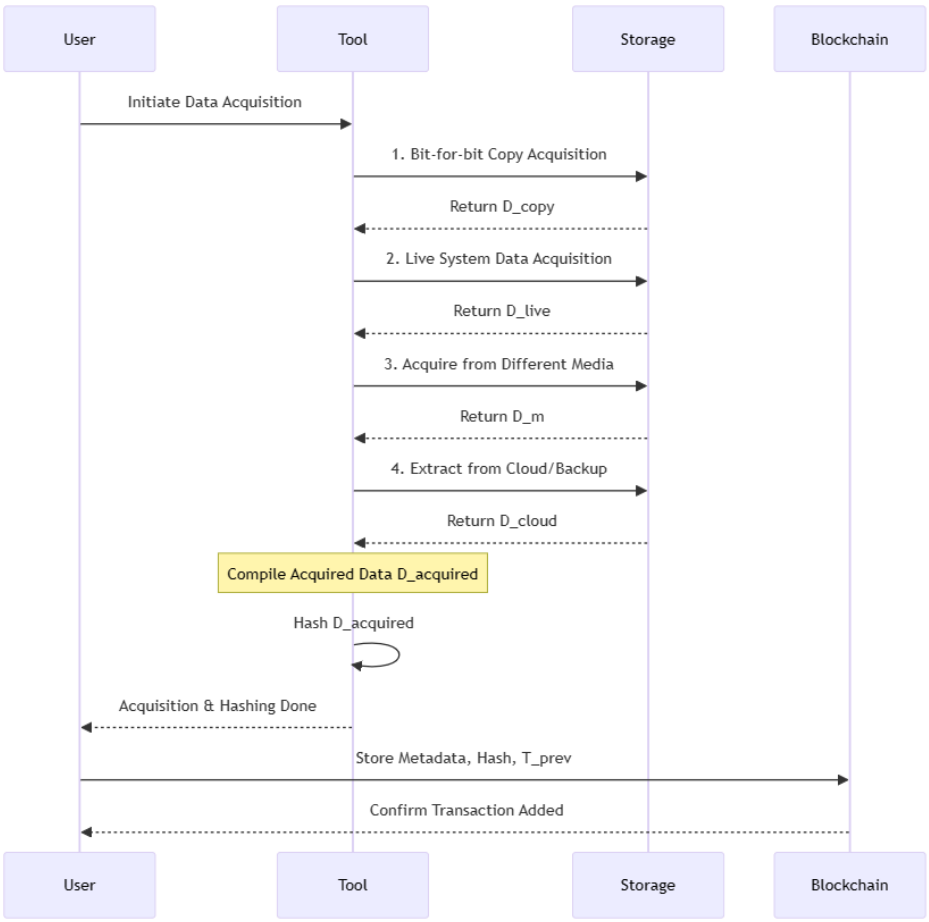

Figure 6. Sequence diagram for data acquisition.

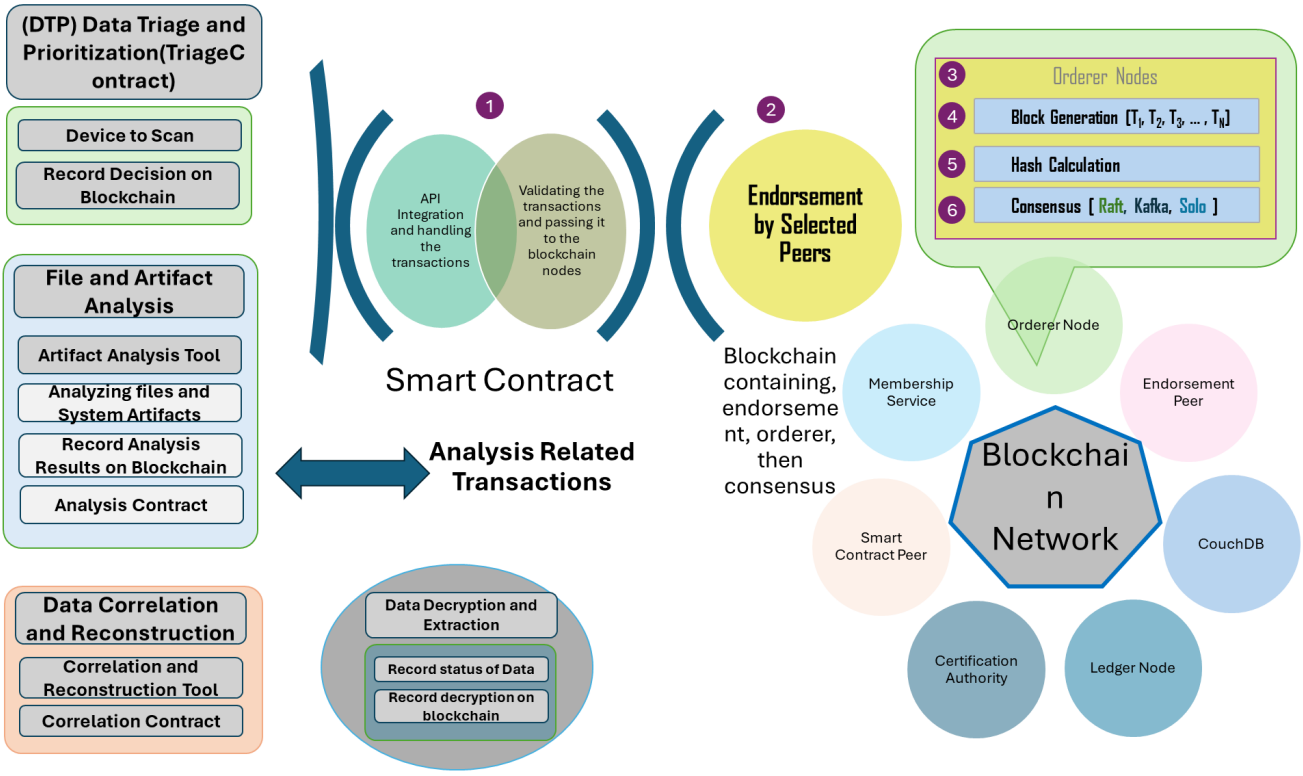

Figure 7. Diagram for integration of data analysis with blockchain.

3.3.1. Formal Definition for Analysis Related Transaction Monitoring

The following formal framework outlines a systematic approach for integrating forensic analysis with blockchain technology. In forensic analysis, evidence (E) is categorized into various properties such as keywords or file types (Ek), recovered files and metadata (Er), logs and system artifacts (El), timeline data (Et), and decrypted or interpreted data (Ed). Each category is assigned a unique hash value using cryptographic functions like SHA-256, ensuring data integrity and immutability. During analysis, each action applied to the evidence, such as search, recovery, or decryption, is recorded as a blockchain transaction (T), encapsulating the hash of the evidence's previous state (Eprev), the action performed (Eaction), the hash of the evidence's current state after the action (Ecurrent), a timestamp, and a cryptographic signature of the analyst. This process not only secures the evidence within a transparent and tamper-proof ledger but also allows for the verification of each forensic step. Anyone can verify the analysis by retrieving the transaction from the blockchain, confirming the authenticity of the signature, and tracing the sequence of actions through the chain of hashes, thus providing a robust framework for forensic integrity and accountability.

Analysis Transactions Monitoring Framework

Let
E represent a forensic evidence entity. It has the following properties:
Ek—keywords or file types being searched.
Er—recovered files and their metadata.
El—logs, system artifacts, and application data.
Et—timeline data.
Ed—decrypted, decompressed, or interpreted data.
Each of these properties can be represented as a unique hash value (using cryptographic hash functions like SHA-256).
Blockchain Integration:
Each step in the forensic analysis will result in a blockchain transaction.
A transaction T can be represented as T=(Eprev, Eaction, Ecurrent, timestamp, signature), where Eprev is the hash of the previous state of the evidence; Eaction is the specific action taken (search, recovery, review, timeline analysis, decryption); Ecurrent is the hash of the current state of the evidence after the action; timestamp is the exact time the action was taken; and signature is a cryptographic signature of the entity performing the action, proving its identity.
Storing Transactions on Blockchain:
For every action applied to forensic evidence, compute Ecurrent by hashing the result of the forensic action; create a transaction T as described above; sign the transaction using the private key of the entity performing the analysis; and add T to the blockchain.

3.3.2. Example Process Flow

Searching for keywords or file types.

Action: Hash the keywords or file types being searched.

Store: Add a transaction with the action as "search" and the details of the search criteria.

Recovering deleted files and analyzing file metadata.

Action: Hash the recovered files and metadata.

Store: Add a transaction with the action as "recover" and the details of the recovered files.

Reviewing logs, system artifacts, and application data.

Action: Hash the logs, artifacts, and data. **Store:** Add a transaction with the action as 'review' and the details of the reviewed data.

Performing timeline analysis.

Action: Hash the timeline data.

Store: Add a transaction with the action as 'timeline analysis' and the details of the analysis.

Using tools and techniques to decrypt, decompress, or interpret the data.

Action: Hash the decrypted, decompressed, or interpreted data.

Store: Add a transaction with the action as 'interpret' and the details of the interpreted data.

The sequence diagram depicted in Figure 8 illustrates the process of integrating forensic data analysis with blockchain storage. The process is initiated when an analyst begins an analysis using a dedicated forensic tool. This tool, upon activation, prompts the analyst to specify the desired forensic action—be it searching for specific keywords, recovering files, or any other pertinent action.

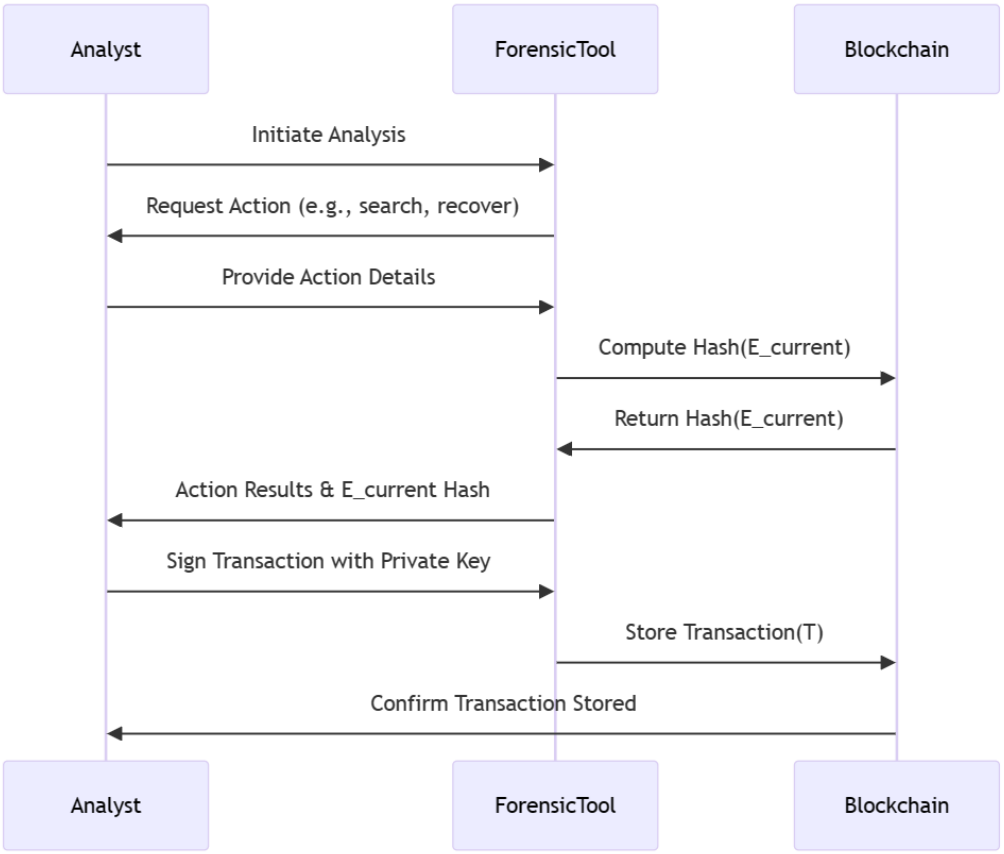

Figure 8. Sequence diagram for data analysis.

3.4. Documentation and Recording

Documentation/reporting provides a verifiable record of the final report and any subsequent updates or amendments. First, a cryptographic hash of the final report is generated. Then, the hash, along with a summary of the report and a timestamp, is added to the blockchain. If the report is updated or amended, the same process is followed to add the new version to the blockchain. This allows for the verification of the original report and any changes made over time.

3.4.1. Formal Definition of Documentation Related Transactions

This formal language framework describes the integration of forensic documentation with blockchain to ensure the integrity and transparency of evidence. Each piece of evidence is encapsulated in a structured data format comprising reports, timelines, visual aids, and expert testimonies, which is then hashed for security. The resultant hash, along with the sender and receiver's addresses, a digital signature, and a timestamp, forms a blockchain transaction. This transaction is added to the blockchain, creating an immutable record of the evidence.

Documentation Transactions Monitoring Framework

Let us denote each of the four steps as follows:
R—report of processes and findings.
T—timeline or chronology.
V—visual aids.
E—expert testimony.
Data Structuring: For each piece of evidence i, we have the following data structure:
Evidencei=R, Ti, Vi, Ei.
Hashing the Evidence: Before adding the evidence to the blockchain, it needs to be hashed to maintain integrity and authenticity. Let us use a cryptographic hash function H():
Hashi=H(Evidencei).
Creating the Blockchain Transaction:
A blockchain transaction typically has the following fields:
Sender's address; receiver's address (in our case, this could be the forensic department's public key); transaction data; signature; and timestamp.
Transactioni=Sender, Receiver, Hashi, Signaturei, Timestampi
Adding to the Blockchain:
Once the transaction is verified, it is added to a block. Blocks are then added to the blockchain in a linear, chronological order.

3.4.2. Integration Model

Forensic experts create evidence documentation (R,T,V,E). The evidence data structure is created and then hashed. The hash, along with other transaction data, is prepared and signed. The transaction is broadcasted to the blockchain network. Network participants (consensus nodes) verify the transaction. Once verified, the transaction is added to a block. The block, when completed, is added to the blockchain.

The sequence diagram depicted in Figure 9 describes the integration of forensic data documentation with blockchain.

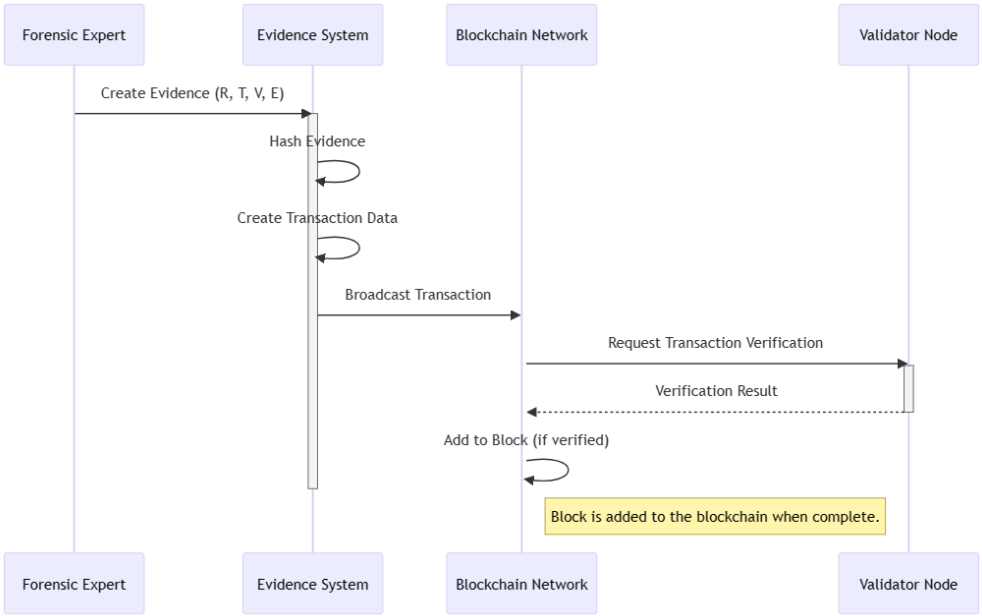

Figure 9. Sequence diagram for data documentation.

The process begins with a forensic expert who interacts with an evidence system to create and document evidence, which encompasses the four steps of report, timeline,

visual aids, and expert testimony, labeled as R, T, V, and E, respectively. The evidence system then takes over. Inside the system, the evidence is first hashed to ensure its integrity and authenticity. Following the hashing, the system packages the hashed evidence into a blockchain transaction format. This transaction, representing the forensic evidence, is then broadcasted to the blockchain network for verification and inclusion. The blockchain network, upon receiving the transaction, solicits the expertise of validator nodes to verify the authenticity and correctness of the transaction. Once a validator node verifies the transaction, it sends back a verification result to the broader blockchain network. If the transaction passes the verification checks, the blockchain network then prepares to include the evidence transaction into a new block. The diagram concludes with a note indicating that once the block is filled with enough verified transactions (or after a certain time), it is finalized and added chronologically to the blockchain. This ensures the permanence and tamper-proofing of the forensic documentation, as each block in the blockchain is immutable and secured through cryptographic measures.

4. Implementation

4.1. Implementation of Data Preservation

In the implementation phase of forensic evidence *preservation* using Hyperledger Fabric, we develop a smart contract, known as the "chaincode", specifically designed for this purpose. The core of this implementation is the *ForensicEvidence* structure, which is intended to represent and store forensic data on the blockchain ledger. The *ForensicContract* struct, embodying the chaincode, facilitates three critical functions. The *InitLedger* function initializes the ledger with its genesis block, setting the foundation for evidence storage. The *AddEvidence* function permits users to add new forensic evidence to the ledger, where each piece of evidence is uniquely identified by an evidence ID and its data are stored in a serialized JSON format for consistency and ease of access. Furthermore, the *GetEvidence* function enables the retrieval of stored forensic evidence using the specific evidence ID. The deployment of this chaincode is managed by the main function, which sets up and starts the chaincode, integrating it into the Hyperledger Fabric network and thus making it operational for forensic evidence preservation and access.

This chaincode (Listing 1) allows entities within the network to add and retrieve forensic evidence, ensuring the preservation steps are recorded immutably on the ledger. Given the permissioned nature of Hyperledger Fabric, only authorized participants can interact with this chaincode, further bolstering the security and integrity of the evidence preservation process. For brevity, only a selected part of the code is included in the following snippet.

Listing 1. Code Snippet of Preservation Related transaction integration with Blockchain.

```
1
2   type ForensicEvidence struct {
3     ForensicImage     string    `json:"forensicImage"`
4     IsolationStatus   bool      `json:"isolationStatus"`
5     DeviceState       string    `json:"deviceState"`
6     ChainOfCustody    []string  `json:"chainOfCustody"`
7     Timestamp         time.Time `json:"timestamp"`
8   }
9
10  type ForensicContract struct {
11    contractapi.Contract
12  }
13
14  func (c *ForensicContract) InitLedger(ctx contractapi.
      TransactionContextInterface) error {
15    genesisEvidence := ForensicEvidence{
16      ForensicImage:   "genesis_image",
17      IsolationStatus: false,
18      DeviceState:     "genesis_state",
19      ChainOfCustody:  []string{},
20      Timestamp:       time.Now(),
```

```
    }

    evidenceJSON, err := json.Marshal(genesisEvidence)
    if err != nil {
      return fmt.Errorf("failed to marshal evidence: %v", err)
    }

    err = ctx.GetStub().PutState("0", evidenceJSON)
    return err
  }

  func (c *ForensicContract) AddEvidence(ctx contractapi.
    TransactionContextInterface, evidenceID string, forensicImage string,
    isolationStatus bool, deviceState string, chainOfCustody []string) error
    {
    evidence := ForensicEvidence{
      ForensicImage:   forensicImage,
      IsolationStatus: isolationStatus,
      DeviceState:     deviceState,
      ChainOfCustody:  chainOfCustody,
      Timestamp:       time.Now(),
    }

    evidenceJSON, err := json.Marshal(evidence)
    if err != nil {
      return fmt.Errorf("failed to marshal evidence: %v", err)
    }

    return ctx.GetStub().PutState(evidenceID, evidenceJSON)
  }

  func (c *ForensicContract) GetEvidence(ctx contractapi.
    TransactionContextInterface, evidenceID string) (*ForensicEvidence, error
    ) {
    evidenceJSON, err := ctx.GetStub().GetState(evidenceID)
    if err != nil {
      return nil, fmt.Errorf("failed to get evidence: %v", err)
    }
    if evidenceJSON == nil {
      return nil, fmt.Errorf("evidence not found for ID %s", evidenceID)
    }

    var evidence ForensicEvidence
    err = json.Unmarshal(evidenceJSON, &evidence)
    if err != nil {
      return nil, fmt.Errorf("failed to unmarshal evidence: %v", err)
    }

    return &evidence, nil
  }
```

4.2. Implementation of Data Acquisition

The implementation phase of the data acquisition process in computer forensics involves capturing digital evidence from various sources and securely storing it for analysis. Utilizing blockchain technology, specifically Hyperledger Fabric, enhances the integrity and verifiability of the forensic data through immutable records.

In this phase, the 'AcquisitionData' structure is designed to encapsulate forensic data acquired from different sources. The implementation includes three key functions: *InitLedger*, a placeholder for initialization logic; *AcquireData*, which processes the unique device ID and acquired data by hashing the latter for integrity and storing them on the ledger associated with the device ID; and *GetData*, enabling the retrieval of acquisition data via the device ID. The main function's role is to initialize and start the smart contract,

laying the groundwork for a blockchain-based forensic data acquisition system. For brevity, we removed these functions from the code below (Listing 2). This setup not only secures forensic data but also provides a scalable framework that can be further developed with additional features and forensic-specific validations, ensuring a robust environment for handling and analyzing digital evidence.

Listing 2. Code Snippet of Smart Contract for Acquisition Related transaction integration with Blockchain.

```
1
2
3   func (fc *ForensicContract) AcquireData(ctx contractapi.
      TransactionContextInterface, deviceID string, acquisitionData
      AcquisitionData) (string, error) {
4     dataBytes := []byte(acquisitionData.DeviceCopyData + acquisitionData.
      LiveSystemData + strings.Join(acquisitionData.StorageMediaData, "") +
      acquisitionData.CloudData)
5     hash := sha256.Sum256(dataBytes)
6     hashString := hex.EncodeToString(hash[:])
7
8     err := ctx.GetStub().PutState(deviceID, dataBytes)
9     if err != nil {
10      return "", fmt.Errorf("Failed to acquire and store data: %v", err)
11    }
12
13    return hashString, nil
14  }
15
16  func (fc *ForensicContract) GetData(ctx contractapi.
      TransactionContextInterface, deviceID string) (AcquisitionData, error) {
17    dataBytes, err := ctx.GetStub().GetState(deviceID)
18    if err != nil {
19      return AcquisitionData{}, fmt.Errorf("Failed to read from world state:
      %v", err)
20    }
21    if dataBytes == nil {
22      return AcquisitionData{}, fmt.Errorf("The device %s does not exist",
      deviceID)
23    }
24
25    var acquisitionData AcquisitionData
26    err = json.Unmarshal(dataBytes, &acquisitionData)
27    if err != nil {
28      return AcquisitionData{}, fmt.Errorf("Failed to unmarshal acquisition
      data: %v", err)
29    }
30
31    return acquisitionData, nil
32  }
33
34
35
```

4.3. Implementation of Data Analysis

The following smart contract (Listing 3) is designed to facilitate forensic transactions on a blockchain, encapsulated through a *ForensicAnalysis* struct. It is operationalized via the *ForensicContract*, a contract that encompasses methods crucial for the operational integrity of the forensic analysis process. These methods include *InitLedger*, which seeds the ledger with a foundational transaction; *AddAnalysis*, for appending new forensic analysis transactions with their data hashed for integrity; and *GetAnalysis*, enabling the retrieval of specific forensic analyses by their ID. This framework serves as a foundational blueprint, necessitating further customization to meet specific operational demands, enhance error handling, and ensure seamless integration within the broader Hyperledger Fabric ecosystem.

Listing 3. Code Snippet of Smart Contract for Analysis Related transaction integration with Blockchain.

```
1
2     // ForensicAnalysis represents a forensic transaction on the blockchain
3     type ForensicAnalysis struct {
4       ID           string `json:"id"`
5       PrevHash     []byte `json:"prevHash"`
6       Action       string `json:"action"`
7       CurrentHash  []byte `json:"currentHash"`
8       Timestamp    string `json:"timestamp"`
9       Signature    string `json:"signature"`
10    }
11
12    // ForensicContract provides functions for managing the forensic analysis
13    type ForensicContract struct {
14      contractapi.Contract
15    }
16
17
18    // AddAnalysis adds a new forensic analysis transaction to the ledger
19    func (c *ForensicContract) AddAnalysis(ctx contractapi.
      TransactionContextInterface, id string, prevHash []byte, action string,
      currentData []byte, signature string) error {
20      hash := sha256.New()
21      hash.Write(currentData)
22      currentHash := hash.Sum(nil)
23
24      analysis := ForensicAnalysis{
25        ID:           id,
26        PrevHash:     prevHash,
27        Action:       action,
28        CurrentHash:  currentHash,
29        Timestamp:    time.Now().String(),
30        Signature:    signature,
31      }
32
33      analysisJSON, err := json.Marshal(analysis)
34      if err != nil {
35        return err
36      }
37
38      return ctx.GetStub().PutState(id, analysisJSON)
39    }
40
41
42
```

4.4. Implementation of Data Documentation

This Go chaincode (Listing 4) lays out a foundational framework aimed at managing forensic documentation within a Hyperledger Fabric blockchain network. It is designed to perform several key functions, including initializing the ledger with a sample forensic document, creating new forensic documents, and enabling the querying of specific forensic documents through their unique IDs. This process facilitates interaction with and the management of forensic documents stored on the blockchain ledger, thereby leveraging blockchain's inherent security and immutability features for enhanced forensic documentation.

Listing 4. Code Snippet of Smart Contract for Documentation Related transaction integration with Blockchain.

```
// ForensicDoc represents the computer forensic documentation steps.
  type ForensicDoc struct {
    Report    string `json:"report"`
    Timeline  string `json:"timeline"`
    VisualAid string `json:"visualAid"`
    Testimony string `json:"testimony"`
  }

  // ForensicContract provides functions for managing the forensic documentation.
  type ForensicContract struct {
    contractapi.Contract
  }


  // CreateForensicDoc adds a new forensic document to the ledger.
  func (fc *ForensicContract) CreateForensicDoc(ctx contractapi.TransactionContextInterface, id string, report, timeline, visualAid, testimony string) error {
    doc := ForensicDoc{
      Report:    report,
      Timeline:  timeline,
      VisualAid: visualAid,
      Testimony: testimony,
    }

    docAsBytes, _ := json.Marshal(doc)
    return ctx.GetStub().PutState(id, docAsBytes)
  }

  // QueryForensicDoc retrieves a specific forensic document from the ledger by its ID.
  func (fc *ForensicContract) QueryForensicDoc(ctx contractapi.TransactionContextInterface, id string) (*ForensicDoc, error) {
    docAsBytes, err := ctx.GetStub().GetState(id)
    if err != nil {
      return nil, fmt.Errorf("Failed to read from world state. %s", err.Error())
    }
    if docAsBytes == nil {
      return nil, fmt.Errorf("The forensic doc %s does not exist", id)
    }

    doc := new(ForensicDoc)
    _ = json.Unmarshal(docAsBytes, doc)

    return doc, nil
  }


```

5. Results

We recently undertook the task of assessing the performance of our Hyperledger Fabric blockchain network, focusing on key metrics such as throughput and latency. To achieve this, we utilized Hyperledger Caliper, a benchmarking tool designed specifically for blockchain networks. Initially, we integrated Caliper with our existing Hyperledger Fabric setup, which was configured on a network of nodes powered by Intel Core i9 processors with 16 GB RAM each. Our Fabric network used the RAFT consensus algorithm and CouchDB for storing transactions. The calibration process involved defining specific benchmark tests in Caliper that mirrored our real-world transaction patterns, which primarily consisted of forensic system transactions like preservation, acquisition, analysis,

and documentation. These benchmarks were then executed against our network, allowing Caliper to systematically submit transactions and record various performance metrics. Through this methodical approach, we were able to measure the transaction throughput (transactions per second) and latency (time taken for transaction confirmation) under different load conditions. These data provided us with valuable insights into the operational efficiency of our blockchain setup and enabled us to optimize our network configurations for better performance.

We successfully set up Hyperledger Caliper to benchmark our Hyperledger Fabric network. The process began with ensuring our Fabric network was fully operational with properly configured peers and orderers. Following this, we installed Hyperledger Caliper using npm, the Node.js package manager, executing 'npm install @hyperledger/caliper-cli'. To align Caliper with our specific version of Fabric, we used the Caliper CLI to bind it, running 'npx caliper bind –caliper-bind-sut fabric:<version>'. The critical part of our setup involved customizing the Caliper benchmark settings and network configuration files. We meticulously defined our blockchain network structure in the network configuration file, detailing peers, orderers, and channel information. In the benchmark file, we outlined our workload, specifying the smart contracts for testing, transaction types, and the rate of transactions. Finally, we executed the benchmark tests with the command 'npx caliper launch manager –caliper-workspace . –caliper-benchconfig <benchmark-config-file> –caliper-networkconfig <network-config-file>'. This allowed Caliper to interact with our Hyperledger Fabric network, process the transactions as defined, and gather crucial performance data, including transaction throughput and latency, providing us with invaluable insights into our network's efficiency and areas for improvement.

The diagram below provides a visual representation of the performance analysis of Hyperledger Fabric, a prominent blockchain framework. At the core of the graph is "Hyperledger Fabric", which branches out into four primary performance metrics: throughput, latency, scalability, and security [69].

1. Throughput is a measure of how many transactions the system can process within a given time frame. In the context of the graph, it emphasizes a high transaction rate, indicating that Hyperledger Fabric is designed to handle a significant volume of transactions efficiently.
2. Latency refers to the time taken to process a single transaction, or the response time. The graph underscores the importance of a low response time, suggesting that transactions in Hyperledger Fabric are processed swiftly.
3. Scalability concerns the system's ability to grow and manage increased demand. The diagram points out that Hyperledger Fabric supports multiple nodes, highlighting its ability to expand and accommodate a growing network.
4. Security is paramount in any blockchain framework. The graph emphasizes robust encryption and privacy features in Hyperledger Fabric, ensuring that data remain secure and private transactions are well protected.

In essence, the diagram (c.f. Figure 10) encapsulates the efficiency, speed, adaptability, and robustness of Hyperledger Fabric in terms of its performance metrics.

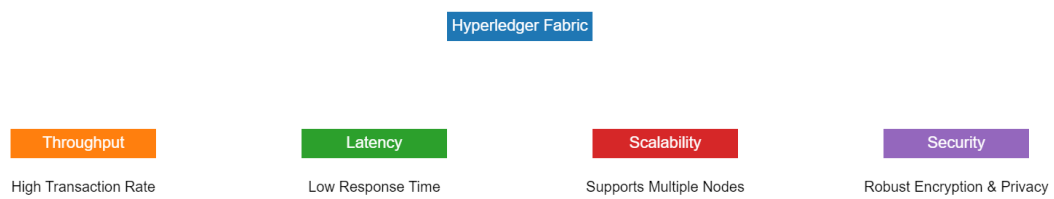

Figure 10. Performance evaluation of Hyperledger Fabric.

Below is a graph that (Figure 11) visualizes the performance metrics for forensic transactions related to preservation, acquisition, analysis, and documentation. The graph shows the number of successful transactions for each type; the throughput measured in transac-

tions per second (TPS); and the latencies (maximum, minimum, and average) experienced during these operations. These visual data can help in understanding the performance and reliability of our Hyperledger Fabric network in handling forensic-related transactions.

Based on the performance metrics provided, we concluded the following.

Preservation (AddEvidence, GetEvidence): The preservation-related transactions showed a high success rate, with 1,100 'AddEvidence' and 28,170 'GetEvidence' transactions completed without failure. The throughput was solid, especially for 'GetEvidence' at 545.5 TPS, surpassing typical Hyperledger Fabric benchmarks, which average around a few hundred TPS under similar conditions. The low average latency of 0.01 s for 'GetEvidence' transactions indicates an exceptionally responsive system. This performance is indicative of an effective preservation system in a forensic context.

Acquisition (AcquireData, GetData): Acquisition transactions also exhibited excellent performance. The 'AcquireData' and 'GetData' operations were successful, with 'GetData' achieving a throughput of 545.5 TPS, on par with the 'GetEvidence' operation, and 'AcquireData' maintaining the network standard at 67.8 TPS. The minimal latency suggests high efficiency, with these metrics showing that the network is optimized for acquisition processes, significantly outperforming average expectations for standard Hyperledger Fabric operations.

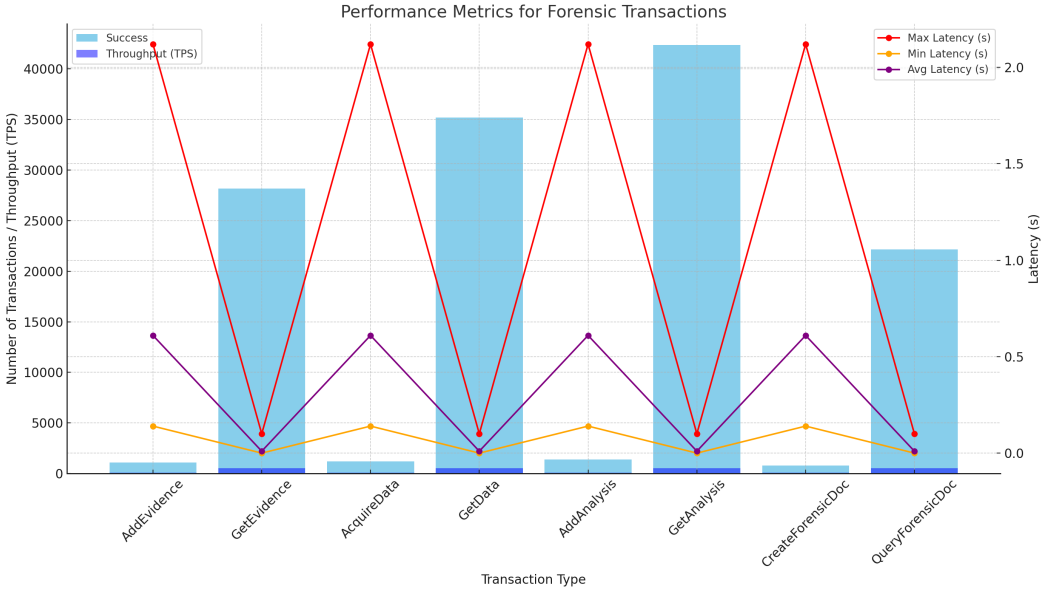

Figure 11. Performance evaluation of Hyperledger Fabric with interims.

Analysis (AddAnalysis, GetAnalysis): For analysis-related transactions, both 'AddAnalysis' and 'GetAnalysis' maintained a 100% success rate, with 'GetAnalysis' showcasing an impressive throughput of 541.3 TPS, which is well above average for typical Hyperledger Fabric networks. The average latency remained low, particularly for 'GetAnalysis', suggesting that the network can handle analysis transactions with high speed and reliability, a crucial aspect for forensic analysis tasks.

Documentation (CreateForensicDoc, QueryForensicDoc): Finally, documentation transactions, crucial for legal integrity, showed strong throughput, with 'QueryForensicDoc' reaching 534.4 TPS. Such a high throughput rate for query transactions is excellent, as it suggests that documentation can be retrieved almost instantaneously, which is beneficial for court proceedings where time is of the essence. The 'CreateForensicDoc' transaction also met the expected standards with a 67.8 TPS throughput and minimal latency.

In conclusion, across all categories—preservation, acquisition, analysis, and documentation—the transactions not only met but exceeded typical performance benchmarks for Hyperledger Fabric networks, indicating a highly efficient and robust system suitable for the demanding requirements of forensic data management.

6. Conclusions

In conclusion, our research marks a pivotal advancement in computer forensics by introducing a generic framework that leverages blockchain technology to embed forensic transactions, enhancing data integrity, authenticity, and transparency throughout the forensic investigation process. By making each forensic action an immutable entry on the blockchain, our approach significantly elevates the standards of reliability and verifiability in the field. The integration of smart contracts offers a seamless interface for forensic activities, ensuring each step in the investigation is recorded in a tamper-proof and transparent manner, which is crucial for legal proceedings. Our empirical evaluation underscored the efficiency and practicality of using smart contracts, highlighting minimal latency and operational overhead. This framework not only bolsters the credibility of forensic evidence in legal contexts but also serves as a versatile model for a broad range of forensic applications, from IoT to cloud computing, contributing profoundly to the integrity of judicial processes and the broader pursuit of justice.

Author Contributions: For this research article, the contributions of the authors were as follows: Conceptualization, methodology, and visualization were solely handled by T.A.S. The remaining tasks, including software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, supervision, project administration, and funding acquisition, were conducted by S.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Business, F. The Compound Annual Growth Rate. 2024. Available online: https://connect.comptia.org/blog/blockchain-statistics (accessed on 25 January 2024).
2. Carter, R. The Ultimate List of Blockchain Statistics. 2023. Available online: https://www.founderjar.com/blockchain-statistics/ (accessed on 16 October 2023).
3. Ali, M.; Ismail, A.; Elgohary, H.; Darwish, S.; Mesbah, S. A Procedure for Tracing Chain of Custody in Digital Image Forensics: A Paradigm Based on Grey Hash and Blockchain. *Symmetry* **2022**, *14*, 334. [CrossRef]
4. Yan, W.; Shen, J.; Cao, Z.; Dong, X. Blockchain based digital evidence chain of custody. In Proceedings of the 2020 The 2nd International Conference on Blockchain Technology, Hilo, HI, USA, 12–14 March 2020; pp. 19–23.
5. Silva, W.; Garcia, A.C.B. Where is our data? A blockchain-based information chain of custody model for privacy improvement. In Proceedings of the 2021 IEEE 24th International Ctoonference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 329–334.
6. Lone, A.H.; Mir, R.N. Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digit. Investig.* **2019**, *28*, 44–55. [CrossRef]
7. Al-Khateeb, H.; Epiphaniou, G.; Daly, H. Blockchain for modern digital forensics: The chain-of-custody as a distributed ledger. In *Blockchain and Clinical Trial: Securing Patient Data*; Springer: Cham, Switzerland, 2019; pp. 149–168.
8. Li, M.; Lal, C.; Conti, M.; Hu, D. LEChain: A blockchain-based lawful evidence management scheme for digital forensics. *Future Gener. Comput. Syst.* **2021**, *115*, 406–420. [CrossRef]
9. Gopalan, S.H.; Suba, S.A.; Ashmithashree, C.; Gayathri, A.; Andrews, V.J. Digital forensics using blockchain. *Int. J. Recent Technol. Eng.* **2019**, *8*, 182–184.
10. Patil, S.; Kadam, S.; Katti, J. Security enhancement of forensic evidences using blockchain. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 263–268.
11. Khan, A.A.; Uddin, M.; Shaikh, A.A.; Laghari, A.A.; Rajput, A.E. MF-ledger: Blockchain hyperledger sawtooth-enabled novel and secure multimedia chain of custody forensic investigation architecture. *IEEE Access* **2021**, *9*, 103637–103650. [CrossRef]
12. Zou, R.; Lv, X.; Wang, B. Blockchain-based photo forensics with permissible transformations. *Comput. Secur.* **2019**, *87*, 101567. [CrossRef]
13. Pocher, N.; Zichichi, M.; Merizzi, F.; Shafiq, M.Z.; Ferretti, S. Detecting anomalous cryptocurrency transactions: An AML/CFT application of machine learning-based forensics. *Electron. Mark.* **2023**, *33*, 37. [CrossRef]

14. Lusetti, M.; Salsi, L.; Dallatana, A. A blockchain based solution for the custody of digital files in forensic medicine. *Forensic Sci. Int. Digit. Investig.* **2020**, *35*, 301017. [CrossRef]
15. Kotsiuba, I.; Velykzhanin, A.; Biloborodov, O.; Skarga-Bandurova, I.; Biloborodova, T.; Yanovich, Y.; Zhygulin, V. Blockchain evolution: From bitcoin to forensic in smart grids. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3100–3106.
16. Brotsis, S.; Kolokotronis, N.; Limniotis, K.; Shiaeles, S.; Kavallieros, D.; Bellini, E.; Pavué, C. Blockchain solutions for forensic evidence preservation in IoT environments. In Proceedings of the 2019 IEEE Conference on Network Softwarization (NetSoft), Paris, France, 24–28 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 110–114.
17. Liao, Z.; Pang, X.; Zhang, J.; Xiong, B.; Wang, J. Blockchain on security and forensics management in edge computing for IoT: A comprehensive survey. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 1159–1175. [CrossRef]
18. Kamal, R.; Hemdan, E.E.D.; El-Fishway, N. A review study on blockchain-based IoT security and forensics. *Multimed. Tools Appl.* **2021**, *80*, 36183–36214. [CrossRef]
19. Li, S.; Qin, T.; Min, G. Blockchain-based digital forensics investigation framework in the internet of things and social systems. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 1433–1441. [CrossRef]
20. Kumar, G.; Saha, R.; Lal, C.; Conti, M. Internet-of-Forensic (IoF): A blockchain based digital forensics framework for IoT applications. *Future Gener. Comput. Syst.* **2021**, *120*, 13–25. [CrossRef]
21. Ryu, J.H.; Sharma, P.K.; Jo, J.H.; Park, J.H. A blockchain-based decentralized efficient investigation framework for IoT digital forensics. *J. Supercomput.* **2019**, *75*, 4372–4387. [CrossRef]
22. Le, D.P.; Meng, H.; Su, L.; Yeo, S.L.; Thing, V. BIFF: A blockchain-based IoT forensics framework with identity privacy. In Proceedings of the TENCON 2018—2018 IEEE Region 10 Conference, Jeju, Republic of Korea, 28–31 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2372–2377.
23. Pourvahab, M.; Ekbatanifard, G. An efficient forensics architecture in software-defined networking-IoT using blockchain technology. *IEEE Access* **2019**, *7*, 99573–99588. [CrossRef]
24. Khanji, S.; Alfandi, O.; Ahmad, L.; Kakkengal, L.; Al-kfairy, M. A systematic analysis on the readiness of blockchain integration in IoT forensics. *Forensic Sci. Int. Digit. Investig.* **2022**, *42*, 301472. [CrossRef]
25. Akinbi, A.; MacDermott, Á.; Ismael, A.M. A systematic literature review of blockchain-based Internet of Things (IoT) forensic investigation process models. *Forensic Sci. Int. Digit. Investig.* **2022**, *42*, 301470. [CrossRef]
26. Mercan, S.; Cebe, M.; Tekiner, E.; Akkaya, K.; Chang, M.; Uluagac, S. A cost-efficient iot forensics framework with blockchain. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
27. Mercan, S.; Cebe, M.; Aygun, R.S.; Akkaya, K.; Toussaint, E.; Danko, D. Blockchain-based video forensics and integrity verification framework for wireless Internet-of-Things devices. *Secur. Priv.* **2021**, *4*, e143. [CrossRef]
28. Sakshi; Malik, A.; Sharma, A.K. Blockchain-based digital chain of custody multimedia evidence preservation framework for internet-of-things. *J. Inf. Secur. Appl.* **2023**, *77*, 103579. [CrossRef]
29. Zhang, Y.; Wu, S.; Jin, B.; Du, J. A blockchain-based process provenance for cloud forensics. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2470–2473.
30. Pourvahab, M.; Ekbatanifard, G. Digital forensics architecture for evidence collection and provenance preservation in iaas cloud environment using sdn and blockchain technology. *IEEE Access* **2019**, *7*, 153349–153364. [CrossRef]
31. Ricci, J.; Baggili, I.; Breitinger, F. Blockchain-based distributed cloud storage digital forensics: Where's the beef? *IEEE Secur. Priv.* **2019**, *17*, 34–42. [CrossRef]
32. Awuson-David, K.; Al-Hadhrami, T.; Alazab, M.; Shah, N.; Shalaginov, A. BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. *Future Gener. Comput. Syst.* **2021**, *122*, 1–13. [CrossRef]
33. Akter, O.; Akther, A.; Uddin, M.A.; Islam, M.M. Cloud forensics: Challenges and blockchain based solutions. *Int. J. Wirel. Microw. Technol.* **2020**, *10*, 1–12. [CrossRef]
34. Ahmed, M.; Reno, S.; Akter, N.; Haque, F. Securing medical forensic system using hyperledger based private blockchain. In Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 19–21 December 2020; IEEE: Piscataway, NJ, USA, 2020, pp. 1–6.
35. Li, M.; Chen, Y.; Lal, C.; Conti, M.; Alazab, M.; Hu, D. Eunomia: Anonymous and secure vehicular digital forensics based on blockchain. *IEEE Trans. Dependable Secur. Comput.* **2021**, *20*, 225–241. [CrossRef]
36. Billard, D. Weighted forensics evidence using blockchain. In Proceedings of the 2018 International Conference on Computing and Data Engineering, Shanghai, China, 4–6 May 2018; pp. 57–61.
37. Mahrous, W.A.; Farouk, M.; Darwish, S.M. An enhanced blockchain-based IoT digital forensics architecture using fuzzy hash. *IEEE Access* **2021**, *9*, 151327–151336. [CrossRef]
38. Lone, A.H.; Mir, R.N. Forensic-chain: Ethereum blockchain based digital forensics chain of custody. *Sci. Pract. Cyber Secur. J.* **2018**, *1*, 21–27.
39. Khan, A.A.; Shaikh, A.A.; Laghari, A.A. IoT with multimedia investigation: A secure process of digital forensics chain-of-custody using blockchain hyperledger sawtooth. *Arab. J. Sci. Eng.* **2023**, *48*, 10173–10188. [CrossRef]

40. Tyagi, R.; Sharma, S.; Mohan, S. Blockchain Enabled Intelligent Digital Forensics System for Autonomous Connected Vehicles. In Proceedings of the 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 10–11 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
41. Malamas, V.; Dasaklis, T.; Kotzanikolaou, P.; Burmester, M.; Katsikas, S. A forensics-by-design management framework for medical devices based on blockchain. In Proceedings of the 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 8–13 July 2019; IEEE: Piscataway, NJ, USA, 2019, Volume 2642, pp. 35–40.
42. Dasaklis, T.K.; Casino, F.; Patsakis, C. Sok: Blockchain solutions for forensics. In *Technology Development for Security Practitioners*; Springer: Cham, Switzerland, 2021; pp. 21–40.
43. Liu, G.; He, J.; Xuan, X. A data preservation method based on blockchain and multidimensional hash for digital forensics. *Complexity* **2021**, *2021*,5536326. [CrossRef]
44. Oladejo, M.T.; Jack, L. Fraud prevention and detection in a blockchain technology environment: Challenges posed to forensic accountants. *Int. J. Econ. Account.* **2020**, *9*, 315–335. [CrossRef]
45. Ahmad, L.; Khanji, S.; Iqbal, F.; Kamoun, F. Blockchain-based chain of custody: Towards real-time tamper-proof evidence management. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual, 25–28 August 2020; pp. 1–8.
46. Nyaletey, E.; Parizi, R.M.; Zhang, Q.; Choo, K.K.R. BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 18–25.
47. Ugwu, M.C.; Okpala, I.U.; Oham, C.I.; Nwakanma, C.I. A tiered blockchain framework for vehicular forensics. *Int. J. Netw. Secur. Its Appl.* **2018**, *10*. Available online: https://aircconline.com/ijnsa/V10N5/10518ijnsa03.pdf (accessed on 1 January 2024).
48. Duy, P.T.; Do Hoang, H.; Hien, D.T.T.; Khanh, N.B.; Pham, V.H. Sdnlog-foren: Ensuring the integrity and tamper resistance of log files for sdn forensics using blockchain. In Proceedings of the 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, 12–13 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 416–421.
49. Srivasthav, D.P.; Maddali, L.P.; Vigneswaran, R. Study of blockchain forensics and analytics tools. In Proceedings of the 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Paris, France, 27–30 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 39–40.
50. Verma, A.; Bhattacharya, P.; Saraswat, D.; Tanwar, S. NyaYa: Blockchain-based electronic law record management scheme for judicial investigations. *J. Inf. Secur. Appl.* **2021**, *63*, 103025. [CrossRef]
51. Casino, F.; Dasaklis, T.K.; Spathoulas, G.P.; Anagnostopoulos, M.; Ghosal, A.; Borocz, I.; Solanas, A.; Conti, M.; Patsakis, C. Research trends, challenges, and emerging topics in digital forensics: A review of reviews. *IEEE Access* **2022**, *10*, 25464–25493. [CrossRef]
52. Amato, F.; Castiglione, A.; Cozzolino, G.; Narducci, F. A semantic-based methodology for digital forensics analysis. *J. Parallel Distrib. Comput.* **2020**, *138*, 172–177. [CrossRef]
53. Dimitriadis, A.; Ivezic, N.; Kulvatunyou, B.; Mavridis, I. D4I-Digital forensics framework for reviewing and investigating cyber attacks. *Array* **2020**, *5*, 100015. [CrossRef]
54. Tully, G.; Cohen, N.; Compton, D.; Davies, G.; Isbell, R.; Watson, T. Quality standards for digital forensics: Learning from experience in England & Wales. *Forensic Sci. Int. Digit. Investig.* **2020**, *32*, 200905.
55. Butt, G.Q.; Sayed, T.A.; Riaz, R.; Rizvi, S.S.; Paul, A. Secure healthcare record sharing mechanism with blockchain. *Appl. Sci.* **2022**, *12*, 2307. [CrossRef]
56. Syed, T.A.; Siddique, M.S.; Nadeem, A.; Alzahrani, A.; Jan, S.; Khattak, M.A.K. A novel blockchain-based framework for vehicle life cycle tracking: An end-to-end solution. *IEEE Access* **2020**, *8*, 111042–111063. [CrossRef]
57. Ali, T.; Nadeem, A.; Alzahrani, A.; Jan, S. A transparent and trusted property registration system on permissioned blockchain. In Proceedings of the 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), Al Madinah Al Munawwarah, Saudi Arabia, 10 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
58. Syed, T.A.; Jan, S.; Siddiqui, M.S.; Alzahrani, A.; Nadeem, A.; Ali, A.; Ullah, A. CAR-tourist: An integrity-preserved collaborative augmented reality framework-tourism as a use-case. *Appl. Sci.* **2022**, *12*, 12022. [CrossRef]
59. Jan, S.; Musa, S.; Ali, T.; Nauman, M.; Anwar, S.; Ali Tanveer, T.; Shah, B. Integrity verification and behavioral classification of a large dataset applications pertaining smart OS via blockchain and generative models. *Expert Syst.* **2021**, *38*, e12611. [CrossRef]
60. Jan, S.; Ali, T.; Alzahrani, A.; Musa, S. Deep convolutional generative adversarial networks for intent-based dynamic behavior capture. *Int. J. Eng. Technol.* **2018**, *7*, 101–103.
61. Ali, S.; Wang, G.; White, B.; Cottrell, R.L. A blockchain-based decentralized data storage and access framework for pinger. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1303–1308.
62. Abdeen, M.A.; Ali, T.; Khan, Y.; Yagoub, M. Fusing identity management, HL7 and Blockchain into a global healthcare record sharing architecture. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 630–636. [CrossRef]
63. Syed, T.A.; Siddiqui, M.S.; Abdullah, H.B.; Jan, S.; Namoun, A.; Alzahrani, A.; Nadeem, A.; Alkhodre, A.B. In-depth review of augmented reality: Tracking technologies, development tools, AR displays, collaborative AR, and security concerns. *Sensors* **2022**, *23*, 146. [CrossRef] [PubMed]

64. Ali, M.S.; Vecchio, M.; Putra, G.D.; Kanhere, S.S.; Antonelli, F. A decentralized peer-to-peer remote health monitoring system. *Sensors* **2020**, *20*, 1656. [CrossRef] [PubMed]
65. Abutaleb, R.A.; Alqahtany, S.S.; Syed, T.A. Integrity and Privacy-Aware, Patient-Centric Health Record Access Control Framework Using a Blockchain. *Appl. Sci.* **2023**, *13*, 1028. [CrossRef]
66. Syed, F.; Gupta, S.K.; Hamood Alsamhi, S.; Rashid, M.; Liu, X. A survey on recent optimal techniques for securing unmanned aerial vehicles applications. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4133. [CrossRef]
67. Guo, H.; Yu, X. A survey on blockchain technology and its security. *Blockchain Res. Appl.* **2022**, *3*, 100067. [CrossRef]
68. Bhutta, M.N.M.; Khwaja, A.A.; Nadeem, A.; Ahmad, H.F.; Khan, M.K.; Hanif, M.A.; Song, H.; Alshamari, M.; Cao, Y. A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access* **2021**, *9*, 61048–61073. [CrossRef]
69. Nasir, Q.; Qasse, I.A.; Abu Talib, M.; Nassif, A.B. Performance analysis of hyperledger fabric platforms. *Secur. Commun. Netw.* **2018**, *2018*, 3976093. [CrossRef]