symmetry

MDPI

Article

A Procedure for Tracing Chain of Custody in Digital Image Forensics: A Paradigm Based on Grey Hash and Blockchain

Mohamed Ali [1], Ahmed Ismail [1], Hany Elgohary [2], Saad Darwish [3,*] and Saleh Mesbah [4]

[1] Department of Forensic Medicine, Ministry of Justice, Higher Institute for Tourism, Hotels and Computer, Al-Seyouf, Alexandria 21533, Egypt; admin@onehoster.com (M.A.); Gisapp13@gmail.com (A.I.)
[2] Expert Counterfeiting and Forgery Research, Department of Forensic Medicine, Ministry of Justice, Alexandria 21519, Egypt; hany_elgohary44@yahoo.com
[3] Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21526, Egypt
[4] College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 1029, Egypt; saleh.mesbah@aast.edu
* Correspondence: saad.darwish@alexu.edu.eg; Tel.: +20-01222632369

Abstract: Digital evidence is critical in cybercrime investigations because it is used to connect individuals to illegal activity. Digital evidence is complicated, diffuse, volatile, and easily altered, and as such, it must be protected. The Chain of Custody (CoC) is a critical component of the digital evidence procedure. The aim of the CoC is to demonstrate that the evidence has not been tampered with at any point throughout the investigation. Because the uncertainty associated with digital evidence is not being assessed at the moment, it is impossible to determine the trustworthiness of CoC. As scientists, forensic examiners have a responsibility to reverse this tendency and officially confront the uncertainty inherent in any evidence upon which they base their judgments. To address these issues, this article proposes a new paradigm for ensuring the integrity of digital evidence (CoC documents). The new paradigm employs fuzzy hash within blockchain data structure to handle uncertainty introduced by error-prone tools when dealing with CoC documents. Traditional hashing techniques are designed to be sensitive to small input modifications and can only determine if the inputs are exactly the same or not. By comparing the similarity of two images, fuzzy hash functions can determine how different they are. With the symmetry idea at its core, the suggested framework effectively deals with random parameter probabilities, as shown in the development of the fuzzy hash segmentation function. We provide a case study for image forensics to illustrate the usefulness of this framework in introducing forensic preparedness to computer systems and enabling a more effective digital investigation procedure.

Keywords: blockchain; chain of custody; digital evidence; digital forensics; fuzzy hash; image forensic

check for updates

Citation: 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. https://doi.org/10.3390/sym14020334

Academic Editors: Ming-Chin Chuang and Chin-Ling Chen

Received: 8 December 2021
Accepted: 27 January 2022
Published: 6 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Any digital data containing trustworthy information that supports an event hypothesis is considered digital evidence. Digital evidence's extent is continuously increasing, including both established and emerging technology such as computers, networks, memory, and mobile devices [1]. Digital evidence has many features, including the ease with which it can be copied and transferred, the ease with which it can be changed and deleted, the ease with which it may be tainted by new data, and the fact that it is time-sensitive. Additionally, digital evidence may be easily transferred across countries. As a result, managing digital evidence is much more complex than processing physical evidence [2]. Digital evidence may take the form of images, videos, text, or device logs. Additionally, it incorporates data from social media platforms such as Twitter, Instagram, and Facebook [3–10].

There are many ways for enhancing the integrity of digital evidence. These techniques include cyclic redundancy checking, hashing functions, digital signatures, time stamps,

encryption, and watermarking. Each technique has a number of benefits and drawbacks; see [8,11–14] for more details. The majority of digital forensic tools and apps use some kind of hashing algorithm to ensure the integrity of digital evidence. Hashing is a cryptographic method for determining an entity's unique representation. When utilizing the conventional hash, certain problems will occur, particularly regarding data integrity since digital data can readily altered. Tampering will always be a problem. This occurs as a result of the exchange procedure being poorly documented [15]. Additionally, a conventional hash cannot be utilized to calculate similarity or to identify traces of evidence. Fuzzy hashing is a kind of hashing that is used to determine the degree to which two entities are similar. Fuzzy hashing enables the investigator to concentrate on possibly incriminating images that would not be seen using conventional hashing techniques.

Meanwhile, a Chain of Custody (CoC) is a critical process in the management of evidence and investigations. CoC is a term that refers to the process of preserving and documenting the chronological history of digital evidence [4–6]. CoC and integrity of digital evidence play a part in the digital process of forensic investigation since forensic investigators must know where, when, and how digital evidence was found, gathered, tracked, handled, and preserved throughout its trip to a court of law. A proper CoC must include documentation that addresses each of these points. If any one of these questions is left unanswered, the CoC is compromised and disturbed. Without a certificate of conformity, the evidence is useless [7–15].

There are many indications that may be used to identify problems with the management of CoC [6,16–19]: (1) threats to the data integrity of digital evidence throughout its lifetime; (2) a massive amount of data is produced by billions of linked devices and must be stored, presenting significant difficulties in ensuring authenticity; (3) because digital evidence is complicated and volatile, and may be altered inadvertently or incorrectly after acquisition, the CoC must guarantee that the evidence gathered is admissible in court; (4) as the number of devices and types of software in the computer and information technology fields continues to increase, cybercrime faces difficulties in terms of the amount of evidence being examined; (5) documentation of the CoC is secure. This is a critical problem since digital evidence may be copied and transferred to other systems; and (6) CoC adaptability and capacity, which comes as a result of the growing amount of data produced by different new digital forensics technologies.

To address the aforementioned issues, an integrated system is required. This system must be capable of presenting data with established integrity and storing CoC for digital evidence, providing an auditing facility to ensure the accuracy of forensic tools and their application procedures, and preserving the artifacts of the evidence, in order for digital evidence to be admissible in court [6,15]. The blockchain may be used to verify the validity and legality of the processes used to collect, store, and transmit digital evidence, as well as to offer a consolidated view of all CoC interactions [20].

In its simplest form, a blockchain is a collection of linked data structures called blocks that store or monitor the state of any distributed system on a peer-to-peer network. Each block is connected to the previous block via a special pointer called a hash pointer, resulting in an append-only system, a permanent and irreversible history that can be used as a real-time audit trail by any participant to verify the accuracy of the records simply by reviewing the data itself [9]. The blockchain has been extensively utilized in a variety of areas, including cloud security, IoT security, and digital forensics. Blockchain technology is also a potential method for evidence verification and management in the area of digital forensics, and it is being extensively explored [10].

Digital image forgeries are becoming more prevalent today since image manipulation software is widely accessible and the usage of digital images has grown in popularity. One cannot tell if the image is genuine or has been altered. Images may be altered by removing a portion of the image, hiding an area within the image or altering the image in such a way that the image information is misrepresented. These flaws erode the validity of digital images [4]. Numerous methods are discussed in detail in order to identify image forgery.

They are categorized as active or passive algorithms [5]. The active method involves embedding a watermark into the picture. Because embedding watermarks in images needs specially equipped cameras, this technique is very restricted in practice. In contrast, passive methods to forgery detection rely on the evidence left on the image by various processing stages during image modification. Passive may also be used to detect the amount and location of forgeries in an image.

To summarize, computer forensics professionals use forensic software to acquire copies or images of electronic equipment and to capture associated data. Recent advances in forensic software allow for remote gathering and analysis. Even if it is impossible to precisely quantify the uncertainty inherent in a piece of digital evidence, courts should consult experts to get a sense of the data's reliability. Every piece of digital data has some degree of uncertainty, and an expert should be able to describe and estimate the degree of certainty that can be put on a particular piece of evidence. If we do not attempt to quantify uncertainty in digital evidence, one might argue that there is no foundation for assessing the evidence's dependability or correctness. Additionally, forensic examiners who do not account for ambiguity throughout their analysis risk arriving at incorrect conclusions during the investigation stage and finding it more difficult to defend their claims when cross-examined.

This paper focuses on the research of protecting digital evidence that is uncertain, which is still a challenging research topic that has not been studied much by researchers. Traditional blockchain-based chain of custody is mainly based on a concise description of the evidence under examination and some kind of hash code. However, the conventional hash method is inefficient at dealing with identical files that arise from benign or malicious alteration of the images examined by the forensic investigator. Utilizing fuzzy hash functions enables forensic investigators to successfully deal with permissible alteration to digital evidence, while using conventional hash methods is ineffective in this situation.

The remainder of this paper will be structured as follows. Section 2 discusses several similar works and their benefits and drawbacks. The suggested framework is described in Section 3. Section 4 outlines the experiments used to verify the proposed framework, and Section 5 concludes the paper.

2. Literature Review

Numerous methods have been presented to enhance the quality of CoC. Several blockchain-based secure digital evidence systems have been suggested in recent years. The authors in [21] suggested a Blockchain-based Chain of Custody (B-CoC) to dematerialize the CoC procedure while ensuring the integrity of gathered evidence and owner traceability. B-CoC was shown to effectively assist the CoC process during the performance assessment. However, the degree of anonymity for validators must be increased without modifying security attributes. In a similar manner, the authors in [15] integrated the Digital Evidence Cabinet (DEC) architecture with Blockchain. This prototype is referred as (B-DEC). B-DEC makes use of data storage integrity to handle digital evidence that relates to DEC. DEC is written in an XML format. However, the system must be capable of securely storing digital evidence through software. It needs to significantly strengthen the protection of digital evidence, such as via the use of encryption.

The work in [8] established a reliable time-stamping technique for protecting digital evidence during the investigative process. Timestamps are acquired from a secure third party in order to establish the date and time of the staff's access to the evidence. A significant issue here is that a reliable source of time is contingent on the setting of the clock that produces it. Another similar study is [12], in which the authors utilized a variety of security techniques to protect the integrity of the digital evidence, including (CRC—Hash Functions—Digital Signatures). SHA512 was chosen for integrity protection based on tests and evaluations since it is computationally extremely fast and least susceptible. However, one may alter the original data, recalculate the hash, and then exchange the original hash with the recalculated one, thus subverting the integrity service.

The authors in [19] encrypted the XML structure on the digital chain of custody data storage using the RC4 cryptography technique. One benefit of utilizing XML is that it is simple for non-professionals to comprehend. Another issue is that XML does not need a specific database management system to be opened. On the other hand, since the material is accessible to everyone, the integrity of digital evidence cannot be accepted in court. Additionally, RC4 encryption will take longer if the plaintext is lengthy. The researchers in [22] evaluated two automated disk imaging programs (Encase and FTK Imager). These programs claim that they protect the integrity of digital evidence by computing MD5 and SHA1 hashes of extracted data. The offered solution is both effective and practical. However, MD5 and SHA1 hashes are insufficient to ensure the evidence's integrity.

Z. Tian et al. [10] proposed a secure Digital Evidence Framework (Block-DEF) based on Blockchain technology, with a loose coupling structure in which evidence and evidence information are stored independently. The Blockchain is used to keep just the evidence information, which is then kept on a trustworthy storage platform. Experiments demonstrated that Block-DEF is a scalable framework; it ensures the authenticity of evidence and strikes an appropriate balance between privacy and traceability. However, when adding a new node to the blockchain it takes an inordinate amount of time to download and validate the blockchain.

While earlier blockchain-based image forensics systems employed standard hashing, the suggested approach uses fuzzy hashing to examine the blockchain validity (evidence items) in order to better handle evidence item alterations induced by both benign and malicious cyberattacks. When the similarity between two blocks surpasses 95%, the block is considered to be original.

3. Methodology

This section explains the suggested methodology for integrating digital evidence in the presence of certain defects (uncertainty of integrity) for many versions of the same document. The phase of data gathering encompasses all image forensic-capture methods. To maintain CoC throughout this phase, the examiner must adhere to forensic standards while acquiring data sources (e.g., hard drives, network packet captures, OS and application logs, memory contents, and mobile devices). With respect to the CoC, blockchain technology, especially when combined with fuzzy hashing, has the potential to provide tamper-proof recording of evidence. By using fuzzy hash functions, forensic investigators may effectively address permitted modification of digital evidence, while traditional hash techniques are useless in this scenario. The suggested framework's fundamental process is shown in Figure 1. Each stage will be discussed in depth.

The efficiency of the proposed system has been verified for application in the field of image forensics. Only images are used in the paper. However, this is a universal approach for different types of data such as audio, video, image, and files. The reasons for choosing images in the application lie in the following factors: (1) a large number of cases within the scope of the work of digital forensics experts are related to image counterfeiting as they represent the main segment in transactions for individuals, such as images of signatures and checks; (2) with the advancement and availability of powerful image processing software tools and computer technology, it is very easy to manipulate digital images. So, it is essential to determine the authenticity, integrity, reliability, and origin of digital images; (3) images can be used in very important fields such as forensic science, medicine, astronomy, and surveillance.

The investigator does not modify the evidence, but the evidence may be altered by benign modification within some application such as compression. The pseudo-randomness of cryptographic hash algorithms makes it hard to identify similar files even if one bit of the input is changed. A hash function that does not retain the resemblance of files (e.g., different versions of a file) is necessary in the area of computer forensics. How forensic investigators may use traces from such situations is becoming more difficult to determine.

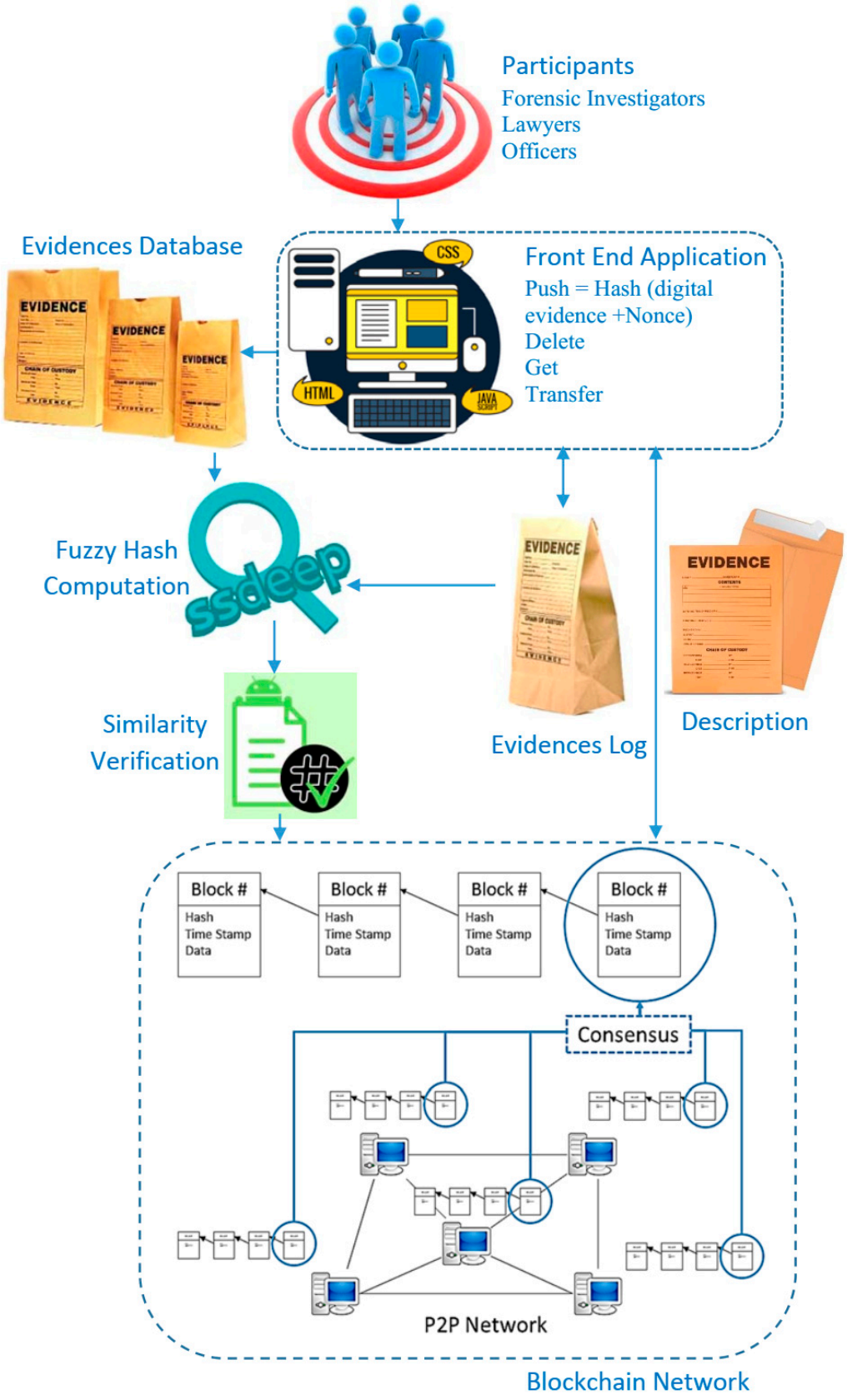

Figure 1. Proposed framework for protecting digital evidence integrity under uncertainty.

3.1. Participants

Authorized parties (forensic investigation) gather digital evidence (images) and then register it in blockchain. Lawyers, the police, the defense, and the court all participate in forensic investigations because they need information regarding the CoC at various points throughout an investigation. Only authorized parties have access to the data associated with a specific piece of evidence [6]. Each authorized entity has a unique identity that is publicly known, and he or she possesses credentials that enable authentication and action throughout the CoC process [21].

3.1.1. Front End

This part is intended to serve as an interface for authorization, to access permissions, and for media. It allows for the downloading of digital evidence and certificates of authenticity, in line with access permissions and levels. The Blockchain interface enables participants to see, invoke, and query blocks, transactions, and chain codes [15,20]. The front end produces a hash of the digital evidence and a nonce that uniquely identifies it (Evidence ID). As the hash generates the ID and the value nonce is randomly selected to guarantee the uniqueness of the evidence's identification, it aids in preserving the integrity of digital evidence throughout its lifetime [21].

3.1.2. Evidence Log

The evidence log keeps track of user interactions with digital evidence. This Evidence Log is implemented on the blockchain and contains information on each piece of evidence on which decision-making depends, including its ID, a description, the submitter's (creator's) identity, and the full history of owners up to the present one, including the dates of ownership transfers. The evidence log is built on top of a peer-to-peer network that includes all authorized entities. A network of this kind may be split into two distinct groups of nodes [15,21]: (1) validator nodes: they are primarily responsible for maintaining a copy of the blockchain; validating transactions; and creating, proposing, and adding blocks to the chain (i.e., participate in the consensus protocol). (2) Lightweight nodes: they are considered clients of the chain since they just issue transactions and depend on validators to add and validate them.

3.2. Blockchain

A blockchain is a decentralized ledger that is maintained by trustless nodes in a peer-to-peer network. Data are stored on the blockchain in blocks that are linked through a connection to the hash value of each block. It is not feasible to modify data in the midst of a block [15]. The first responder initiates forensic-chain by hashing digital evidence (image) and securely storing it on the blockchain through the smart contract. Additional information such as the time and date of the incident, the location of the crime scene, the address to which evidence is transferred, and the present condition of the evidence are also stored on the blockchain. The chain of custody for digital forensics on the blockchain has the potential to significantly improve forensic applications by ensuring the integrity and security of digital evidence while achieving the intended result [9].

As there are just a few peers connected to the network, the block size on the blockchain is smaller. In contrast, not all nodes are required to download the whole blockchain in order to be operational. Some members in the blockchain network participate just for the purpose of making transactions and not for the purpose of verifying them. Full nodes are a subset of nodes that fall into this category. Participation in the current transaction requires the use of complete nodes. The block headers and transactions in each block must be downloaded in their entirety, which implies users must download the whole blockchain's contents. With the ever-increasing size of the blockchain, scalability also becomes a problem. Furthermore, the blockchains' number is determined by the number of available digital pieces of evidence.

The proposed system depends on the piecewise hashing technology for cryptography since the main contribution is to handle uncertainty in CoC. Piecewise hashing uses an arbitrary hashing algorithm to create many checksums for a file instead of just one. Rather than generating a single hash for the entire file, a hash is generated for many discrete fixed-size segments of the file. The following characteristics describe this particular kind of hashing: (1) a hash function should be computationally difficult to reverse "pre-image resistance"; (2) it should be difficult to discover another input with the same hash if you know the hash of the input you're looking for "second Pre-Image Resistance"; (3) it should be difficult to locate two inputs of the same length that have the same hash value if this characteristic is present "collision-free hash function" [22–24].

3.2.1. Piecewise Hashing

To account for the uncertainty associated with evidence item changes, we utilized Fuzzy Hashing (FH) rather than conventional hashes such as SHA 256 in our study. FH, also known as Context-Triggered Piecewise Hashing (CTPH), is a mix of Piecewise and Rolling Hashing (RH). Unlike traditional hashes, where their hashes (checksums) can be interpreted as correct or incorrect, and as black or white, CTPH is more akin to the "grey hash type" as it can identify two files that are likely near duplicates of one another but would not be detected using traditional hashing methods [23]. RH generates 'segments' of conventional hash strings by generating a pseudo-random value depending on the context of the input. In comparison, PH (Piecewise Hashes), such as conventional hashes, produce a final checksum for the whole picture. They circumvent the latter's restrictions by segmenting the whole image into defined segments and then generating hash values for each of these parts. Finally, the produced values comprise the final hash sequence. FH employs the concept of PH to preserve data similarity in this study. Additionally, PH was designed to minimize possible mistakes during forensic imaging, ensuring that the data's integrity is absolute and complete since only one hash segment is void [23,24].

3.2.2. Approximate Matching

Approximate matching is an exciting new technique for determining the similarities between two digital objects. Numerous approximation matching techniques developed to address contemporary issues in digital forensics are essentially based on the capacity to describe objects as sets of characteristics, which simplifies the similarity problem by limiting it to the well-defined domain of set operations [25]. There are eight well-known approximation matching algorithms, including the following ssdeep, sdhash, mrsh-v2, bbHash, mvHash-B, SimHash, saHash, and TLSH. While the first three algorithms remain expanded and relevant, the last four algorithms are less promising in terms of digital forensics for a variety of reasons, including recall and accuracy rates, runtime efficiency, and detection capabilities. For cross-correlation, the final method (TLSH) is less powerful than sdhash and mrsh-v2 [25]. While ssdeep is the most well-known CTPH use today, another method, Multi-Resolution Similarity Hashing, version 2 (MRSH-V2), has been suggested based on the same principles or enhancements to the original ssdeep algorithm [26].

In ssdeep, the system computes the similarity of two files based on their signatures throughout the comparison process [26]. Ssdeep analyzes two strings and calculates the least number of operations required to convert one string into the other using an edit distance method based on Levenshtein distance. While ssdeep is very efficient at detecting similarities between text files, it has a poor detection rate for images due to the possibility of an active adversary exploiting it [23]. In comparison, Sdhash (Similarity Digest hash) encodes the output hash features with a low empirical probability using Bloom Filters. Its results are based on a "similarity score calculated by computing the normalized entropy of the digests, which runs from 0 to 100, with 0 being a mismatch and 100 representing a perfect or near match. The sole drawback discovered for sdhash was its execution time [23].

Mrsh-v2 overcomes ssdeep's limitations and becomes quicker than sdhash [25]. The main objective of MRSH-v2 is to compress and produce a similarity digest for every byte

sequence. Similarity digests are constructed in such a manner that they may be compared to one another, generating a similarity score. Each digest of similarity is composed of Bloom filters. To generate the similarity digest, MRSH-v2 divides the input into roughly 160-byte pieces (sub hashes). These chunks are hashed using FNV (Fowler/Noll/Vo) Fast non-cryptographic hash function to establish the Bloom filter's five bits. To chunk the input, it employs a seven-byte window that glides across it byte by byte. Approximate matching is accomplished by comparing similarity digests. A pairwise comparison of two file sets is needed to compare them [27,28].

The root node of a hierarchical Bloom filter tree is a Bloom filter that represents the whole collection. When searching for an image, if a match is discovered at the root of the tree, all of the tree's child nodes may be searched. The method of determining if a file matches a Bloom filter node is identical to that of adding a file to the tree. Rather than putting each hash into the node, the sub hashes are compared to the Bloom filter to see whether they are included inside it. If a node has a certain number of consecutive hashes, it is considered a match [27].

3.2.3. Similarity

A similarity tool's ultimate aim is to function as a drop-in substitute for the crypto hashes used in forensic file practice for file filtering [28]. Approximate matching may be accomplished using two distinct abstractions: byte wise matching and semantic matching. (1) Byte wise matching: this algorithm works at the byte level and accepts only byte sequences as input. Byte wise algorithms serve two primary purposes. A feature extraction function detects and extracts properties from objects in order to compress them for comparative purposes. Then, a similarity function compares these compressed versions in order to provide a normalized match score. Typically, this comparison is made using string formulae such as Hamming and Levenshtein distances [25]. Byte-wise has a number of restrictions, including [25]. (1) It is unable to detect similarities at a higher level of abstraction, for example, semantically. (2) It is unable to properly match two image files that contain the same semantic image but are stored in various file kinds and formats as a result of their differing binary encodings. (3) Due to the absence of a universally accepted definition of similarity, not all types of byte-level similarity are equally useful since certain artifacts (e.g., headers and footers) are trivial and result in false positives.

This research focuses on the second type, (2) semantic matching, which operates on the content visual layer (i.e., digital evidence images) and thus closely resembles human behavior, for example, the similarity of the content of a JPG and a PNG image, despite the fact that the image file types are different. To put it another way, two images are semantically similar if they convey the same information. For instance, a JPG file is semantically equivalent to an exported PNG file containing the same image. Their cryptographic hashes will not be same, but the images will be identical [25]. To compare two hash values, a comparison function is required. The comparison function takes two hash values as input and returns a number between 0 and X, where X is the maximum match score. A score of X indicates that the hash values are identical or nearly similar, implying that the input files are also identical or nearly identical. The similarity score should ideally be between 0 and 100 and expressed as a percentage.

The suggested framework uses MRSH-v2 for creating the digital grey hash for each block within the blockchain network that utilizes the Hierarchical Bloom Filter Tree (HBFT) approach. As stated in [27], HBFT is quite good at detecting files that share at least 40% of their content, and it has excellent recall when dealing with identical sets of data. This means that the HBFT data structure is an effective alternative to all-against-all comparisons while also delivering significant speed benefits. The HBFT approach yielded a recall level of 95% for similar files when using mrsh-v2 as ground truth. Therefore, the proposed framework has considered 95% as the appropriate metric for resemblance [28]. See [29–34] for more information regarding fuzzy hashing techniques.

3.3. Peer to Peer (P2P) Network

P2P is used to create the network architecture and to facilitate communication between the blockchain layer and the rest of the network (responsible for constructing a blockchain for each node in the underlying network). The majority of blockchain schemes use a peer-to-peer network as a blockchain network. This work utilizes a peer-to-peer network to organize nodes, offers peer-to-peer routing, secures the transfer of proof information, and maintains the Blockchain's consensus. Existing peer-to-peer network methods may be utilized directly or modified to build the Blockchain's network [10].

3.4. Consensus Mechanism

The blockchain consensus process selects a node to generate and broadcast the blockchain next block and ensures that each node's blockchain is consistent [10]. A blockchain transaction is verified via the application of a consensus concept. Consensus ensures that each transaction has its own independent witness mechanism. On the blockchain, there are many forms of consensus, including Proof of Work (PoW), Proof of Stack (PoS), and Proof of Authority (PoA). Consensus types vary according to how the blockchain interacts with data storage [15].

With PoW, nodes compete against one another by solving a mathematical problem to confirm transactions and create new blocks. While solving a block is a computationally demanding job, validating it is straightforward. To further incentivize such a system, solving a block also results in the mining of a certain number of bitcoins, which serves as the incentive for block makers (often referred to as miners) [21]. PoW is suitable for permissionless networks, that is, networks in which nodes may join without prior authorization. The primary disadvantage of PoW is its high energy consumption, which also precludes its use in some situations [21]. This has resulted in the study of other types of blockchain consensus, such as PoA. This study focuses on PoA, which is usually used in permissioned networks, i.e., networks in which nodes cannot join and become validators freely. With the PoA, validators must be pre-authorized and their identities must be known. As a consequence, behaving maliciously leads in a loss of personal reputation and, eventually, expulsion from the validator set [21].

3.5. Hyperledger Blockchain Platform

Hyperledger Fabric (HLF) is a blockchain-based system for electronic digital record exchange across several organizations. Recently, several blockchain systems have been created by different businesses, including Ethereum, Corda, and Ripple [35]. The Hyperledger Composer (HLC) is a framework for building blockchain applications that significantly speeds up and simplifies the process of designing blockchain use cases. One of the many benefits of HLC is that it is completely open-source, with an open governance architecture that allows for contributions by anybody [6]. By design, HLC satisfies all of the criteria for developing an automated system that is both robust and secure in its recording of all the information related to the evidence collection process for a specific cyber forensic case. HLC is compatible with and runs on top of the current HLF blockchain architecture and runtime, enabling pluggable blockchain consensus protocols to guarantee that transactions are verified according to the policy established by the designated business network members [6].

The proposed framework in this article is based on HLF and HLC and offers the following major benefits [6,36]: (1) it is distinguished from the others by its usage of the permissioned blockchain idea, in which transaction processing is delegated to a select group of trustworthy network members; (2) as a consequence, the resulting environment is more regulated and predictable than public permissionless blockchains; (3) block generation does not require resource-intensive computations associated with PoW techniques; (4) due to its modular nature, it enables the employment of a variety of methods to achieve agreement among business process participants; and (5) Ethereum is probably not the ideal

cryptocurrency to use for crime-scene investigation. Digital forensic investigations require confidentiality and are conducted by genuine and trustworthy parties.

From a functional standpoint, the HLF network's nodes are classified as follows [36]: (1) clients initiate transactions, participate in their processing, and broadcast transactions to ordering services; (2) peers execute the transaction processing workflow, verify them, and maintain the blockchain registry; the blockchain registry is an append-only data structure that contains a hash chain of all transactions, as well as a concise representation of the latest ledger state; (3) Ordering Service Nodes (OSN) or, simply, orders establish the general order of all transactions in the blockchain using the distributed consensus algorithm; each transaction contains updates to the system's state, the history of which is stored in the blockchain, and cryptographic signatures of endorsing peers; the separation of processing nodes (peers) and transaction order keeps HLF's consensus as modular as feasible and facilitates protocol replacement.

To define business processes within the framework of the (HLF and HLC) platform, a variety of concepts are employed, the most important of which are assets, participants, and network-stored transactions. (1) Assets: anything of value that can be traded or shared via a network is considered an asset. The suggested approach treats digital evidence and the comprehensive information associated with it as an asset that is kept in HLC's asset registry. (2) Collaborators: participants in the forensic chain model are forensic investigators. In HLC, the participant's structure is represented using a file. It is possible to generate new instances of the modeled participant and add them to the participant register.

Additionally, HLC needs blockchain IDs as a form of identification, and an identity registry stores a collection of mappings between identities and participants. At any point in time, admin peers controlled by companies in the hyperledger composer blockchain consortium may add new participants with suitable identity responsibilities to address a specific scenario. Participants may exchange information in a secure manner using the channels available on the (HLF and HLC) platform. (3) Transactions are used to explain the activities that participants may take on assets as they travel through the network. Transactions in the proposed framework either record information about the evidence or the evidence transfer event on the network. See [37–40] for more information regarding hyperledger blockchain platform.

3.6. Evidence Database

The evidence database is a standard database and/or file repository that stores the actual digital evidence together with an identification ID computed from the evidence's hash and a nonce. This database is disseminated and is maintained by a number of reputable organizations (e.g., law, court, officers). Additionally, each access is granted only if the asking organization is allowed to provide it in accordance with its function. There are two reasons for this split (between the Evidence Log and the Evidences database). To begin, evidence may be too big to be kept effectively on the blockchain (for example, a piece of evidence may be a bit-by-bit copy of a storage device of several TBs of capacity). Second, and most crucially, if pieces of evidence are kept on the blockchain, they are accessible to all nodes in the blockchain network, while only authorized nodes should be permitted to collect evidence. As a result, we keep just information on the CoC process and a hash of the evidence in the blockchain, which enables us to check the integrity of pieces of evidence throughout acquisition [21]. See [41–47] for more information about protecting digital evidence integrity and preserving chain of custody.

4. Performance Evaluation and Analysis

Performance is the most desired characteristic of any problem-solving endeavor, and this is also true for Blockchain-based solutions. We utilized Hyperledger Caliper to assess the performance of our prototype. Caliper enables users to benchmark the performance of various blockchain systems against a specified set of use cases and produce reports that include performance metrics such as transactions per second (tps) and transaction latency

(the time elapsed from the issue of the transaction to its inclusion in the blockchain). The experiments were conducted on an Intel Core i7–5500U, 2.4 GHz processor, 8 GB DDR3 RAM laptop, and Windows 10 operating system. The code was written in Python language using Python 3.6 software.

4.1. Performance Analysis

The first set of experiments was implemented to test our prototype using Caliper's 2-organization-1-peer and 3-organization-1-peer network models with four clients in the first round of tests. To ascertain our suggested framework's transactional efficiency, we created a test file that targeted two primary functionalities of our framework, namely, evidence creation and evidence transfer, due to their direct participation in changing the Blockchain state. We conducted ten rounds of testing with varying transaction volumes and send transaction rates. Multiple runs of the test were required to get the average values of performance indicators with a low chance of error. Tables 1 and 2 show the latency and throughput for various rounds of 2-organization-1-peer and 3-organization-1-peer network architectures. The performance assessment results indicate that the prototype's throughput achieves a maximum value and then begins to decrease as the transmit rate increases. The highest throughputs obtained in 2-organization-1-peer and 3-organization-1-peer network architectures are 15 tps and 10 tps, respectively. Additionally, the results indicate that increasing the number of peers reduces the prototype's throughput, which is consistent with the characteristic of Hyperledger-based consortium Blockchains.

Table 1. Performance evaluation results with 2-organization-1-peer network mode.

Round	Send Rate (tps)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (tps)
1	6	0.85	0.70	0.77	5
2	11	1.18	0.74	0.98	9
3	16	1.46	0.49	1.13	13
4	21	2.89	0.61	1.93	14
5	26	4.06	0.84	2.72	14
6	30	5.80	1.05	4.37	15
7	35	7.27	1.32	5.76	15
8	40	21.61	8.36	16.15	8
9	43	11.49	2.49	8.38	15
10	49	13.88	8.57	11.85	13

Table 2. Performance evaluation results with 3-organization-1-peer network framework.

Round	Send Rate (tps)	Max Latency (S)	Min Latency (S)	Avg Latency (S)	Throughput (tps)
1	6	1.24	1.01	1.16	5
2	11	8.32	2.74	6.34	4
3	16	4.60	1.00	3.13	8
4	21	8.42	5.24	7.01	8
5	26	9.56	3.95	7.11	10
6	30	11.62	3.85	9.07	10
7	33	14.16	3.22	10.99	10
8	39	17.16	10.77	14.34	9
9	46	47.84	19.93	34.37	5
10	50	19.35	12.21	16.29	10

The second round of tests assessed the block generator's load, which is used to determine the distribution of blocks generated by each node. This shows if each node in the blockchain network being used has an equal probability of producing blocks. We utilized a 1000-node architecture in the simulator and created 105 blocks sequentially, counting the blocks generated by each node. The cumulative percentage of produced blocks containing

x nodes is shown in Figure 2, where k is the number of node names. The more evenly distributed the load, the more likely the line will be straight. When k equals one, the curve exhibits a sharp rise. The demand on the generator is balanced evenly by increasing the number of node names. The greater the number of node names, the more linear the growth becomes. However, as the number of node names grows, load balancing's growth impact progressively diminishes. By adding a modest number of node names, these block generators may significantly improve load balancing. The number of blocks produced is centered on the mean. In general, when k equals 5, the load balancing impact is satisfactory for the block generator.

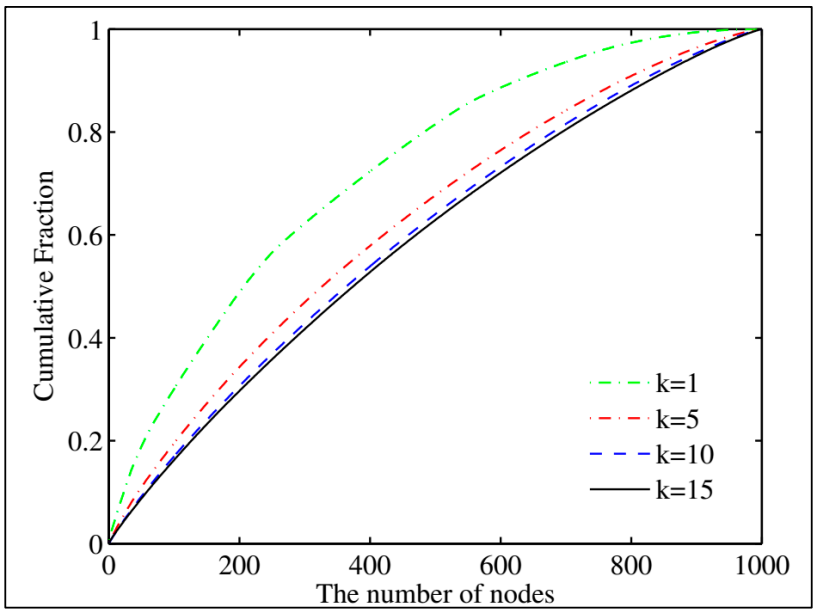

Figure 2. The cumulative distribution of generated blocks.

The third set of experiments was conducted to evaluate the size of the blockchain against different numbers of blocks on a topology with 1000 nodes. The name number was set to one and the group size variable, h, was set to three bits for the topology. A block could contain no more than 2000 transactions. Following that, we determined the blockchain's storage capacity on each node. We were primarily concerned with the distribution of full blocks (block headers and contents) and the blockchain's size. The distribution of full blocks stored by each node represented the blockchain's load balancing mechanism. We conducted the experiment three times. Each time, we adjusted the variable h to create a new group size and then counted the number of full blocks stored in each node. Figure 3 illustrates the blockchain's size as a function of the block count. The maximum, mean, and minimum blockchain sizes are all determined using the mixed blockchain, whereas the entire blockchain size is determined using a typical scenario in which all nodes hold the whole blockchain. The mixed blockchain is much smaller than the regular blockchain. For all four kinds of outcomes, the blockchain's size grows linearly as the number of blocks increases, which is consistent with the theoretical theory.

We conducted the last set of experiments to determine the time required to conduct a full search, in comparison to MRSH-v2, and to determine the approach's success in locating the 100 "illegal" files included verbatim in the hard disk image, as well as the 40 files from the image that are similar to "illegal" files, as defined by MRSH-v2. A collection of simulated "known-illegal" images consisted of 4000 images plus 140 more images, as follows: within the 4000 "illegal" images, there were 100 images; 40 images were not included in the "illegal" images but showed a high degree of resemblance to images in the corpus, as determined by MRSH-v2.

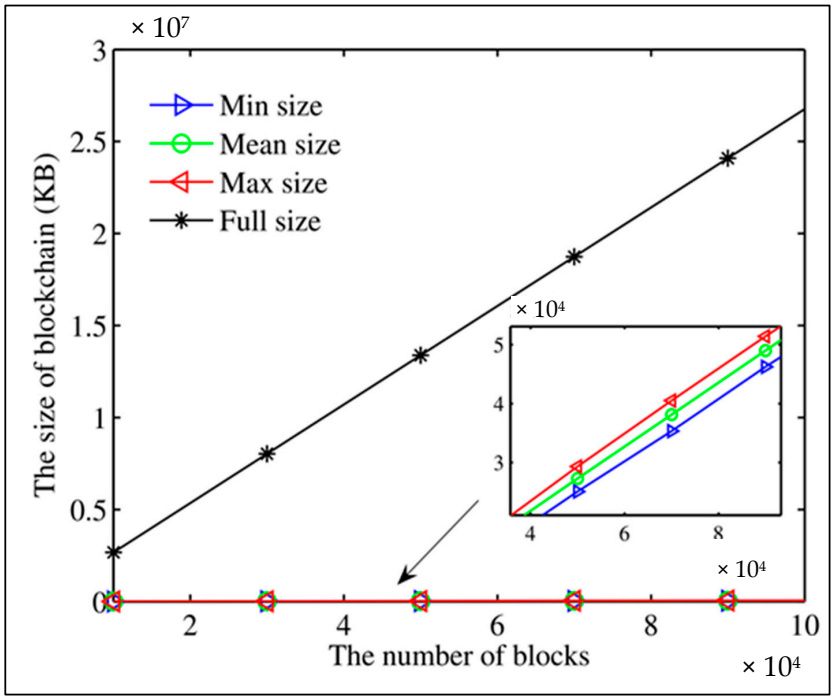

Figure 3. The blockchain's size as a function of the block count.

The main measure was the time needed to execute the whole process, which included the time required to construct the tree, search the tree, and perform pairwise comparisons at the leaves. MRSH-v2 ran for a total of 2592 s. Figure 4 illustrates the running times. The tree was constructed using the smaller sample of 4000 "illegal" images, and then searches were performed for all of the images in the bigger corpus. The "Search Time" column covers both the time spent searching the tree and the time spent doing leaf comparisons. As anticipated, having more leaf nodes resulted in the quickest execution time. The entire duration of the race was 1182 s (a 54% reduction in the time required for an all-against-all pairwise comparison). Due to the paired approach's lack of scalability, this discrepancy is expected to be much more apparent with bigger datasets.

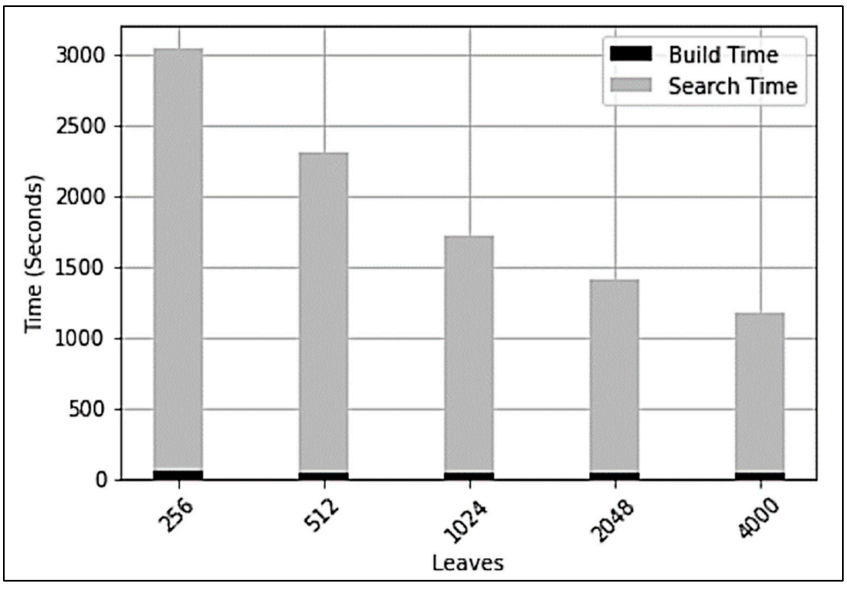

Figure 4. Time to search for planted evidence (including pairwise comparisons).

4.2. Analysis of Possible Attacks

As far as forensics are concerned, both blacklisting and whitelisting attacks are discussed in this section. Anti-blacklisting/anti-whitelisting may be used to conceal information from the perspective of an attacker. An active attacker manipulates a file such that fuzzy hashing does not recognize the files as being identical, which is what is meant by "anti-blacklisting." If a human observer cannot tell the difference between the original and the manipulated version, we consider the attack to be effective. If a file was successfully modified, it would be labelled as an unknown file rather than a known-to-be-bad file. This anti-blacklisting attack aims to alter a single byte inside each chunk while keeping track of the exact locations of the trigger points. Change the triggering such that the extent of each change is determined by the Hamming distance, which is the most apparent concept. As stated in [33], in a worst-case scenario, each building block has a Hamming distance and a 'one-bit change' is enough to manipulate the triggering. In this case, an active adversary approximately needs to change one bit for each position. Actually, a lot of 100 more changes needs to be done as there are also positions where the Hamming distance has small distance.

For anti-whitelisting to work, the attacker must utilize a hash value from one of the files on the whitelist in order to change another file (typically one of the bad ones) such that the new file's hash value is identical to one on the whitelist. An attack is deemed effective if a human observer cannot detect any differences between the original and altered versions. Since files may be created for a given signature by generating legal trigger sequences for each building block and inserting zero-strings in between, this technique is not considered preimage-resistant. Even though it should be feasible, changing the hash value of a particular file will lead to a worthless file. An active adversary's initial action is to delete all currently active trigger sequences. As a second step, he must completely mimic the white-listed file's triggering behavior, which will result in many additional modifications to the system.

5. Conclusions

The integrity and credibility of the digital evidence in a single process for managing the chain of custody are critical components of these operations (or chain of evidence). The purpose of this study is to determine the efficacy of fuzzy hashing algorithms inside blockchain technology, as opposed to conventional cryptographic hash algorithms, in preserving the integrity of digital evidence in image forensics for assessing similarities. We developed and tested a prototype of a forensic chain model based on a hyperledger composer. According to the performance evaluation, fuzzy hash-based blockchains proved to be an effective support for the chain of custody process due to their ability to sustain a realistic workload with a manageable overhead in terms of memory used to store the chain and their ability to handle the chain of custody-related uncertainty. Future work includes testing the efficiency of the suggested framework when handling a large number of digital pieces of evidence.

Author Contributions: Conceptualization, S.D., H.E. and S.M.; methodology, M.A., A.I. and H.E.; software, M.A., A.I. and H.E.; validation, S.D., H.E. and S.M.; formal analysis, S.D., M.A., A.I. and H.E.; investigation, S.D., M.A. and A.I.; resources, S.D., H.E. and S.M.; data curation, M.A., A.I. and H.E.; writing—Original draft preparation, S.D., M.A., A.I. and H.E.; writing—Review and editing, S.D. and S.M., visualization, M.A., A.I. and H.E.; supervision, S.D. and S.M.; project administration, M.A., A.I. and H.E.; funding acquisition, M.A., A.I. and H.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

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

References

1. Dosisa, S.; Homema, I.; Popova, O. Semantic representation and integration of digital evidence. *Procedia Comput. Sci.* **2013**, *22*, 1266–1275. [CrossRef]
2. Prayudi, Y.; Azhari, S. Digital chain of custody: State of the art. *Int. J. Comput. Appl.* **2015**, *114*, 1–9. [CrossRef]
3. Campbell, N.; Goodyear, T.; Messer, W.; Stuart, E.; Fairbanks, J. Digital witness: Remote method for volunteering digital evidence on mobile devices. In Proceedings of the IEEE International Conference on Technologies for Homeland Security, Woburn, MA, USA, 23–24 October 2018; pp. 1–5.
4. Patel, J.; Bhatt, N. Review of digital image forgery detection. *Int. J. Recent Innov. Trends Comput. Commun.* **2017**, *5*, 152–155.
5. Varkey, A.; Nair, L. Robust image forgery detection and classification in copy-move using SVM. *Int. J. Adv. Res. Trends Eng. Technol.* **2018**, *5*, 89–93.
6. Hamid, A.; Naaz, R. Forensic-chain: Blockchain based digital forensics chain of custody with POC in hyperledger composer. *Int. J. Digit. Investig.* **2019**, *28*, 44–55.
7. Sadiku, M.; Shadare, A.; Musa, S. Digital chain of custody. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2017**, *7*, 117–118. [CrossRef]
8. Ćosić, J.; Baca, M. (Im) Proving chain of custody and digital evidence integrity with time stamp. In Proceedings of the IEEE International Conference on Convention, Opatija, Croatia, 21–23 May 2010; pp. 1226–1230.
9. Auqib, L.; Roohie, M. Forensic-chain: Ethereum blockchain based digital forensics chain of custody. *Sci. Pract. Cyber Secur. J.* **2017**, *1*, 21–27.
10. Tian, Z.; Li, M.; Qiu, M.; Sun, Y.; Su, S. Block-DEF: A secure digital evidence framework using blockchain. *Int. J. Inf. Sci.* **2019**, *491*, 151–165. [CrossRef]
11. Ćosić, J.; Ćosić, Z.; Baca, M. An ontological approach to study and manage digital chain of custody of digital evidence. *Int. J. Inf. Organ. Sci.* **2011**, *35*, 1–13.
12. Saleem, S.; Popov, O.; Dahman, R. Evaluation of security methods for ensuring the integrity of digital evidence. In Proceedings of the IEEE International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates, 25–27 April 2011; pp. 220–225.
13. Rasjid, Z.; Soewito, B.; Witjaksono, G.; Edi, A. A review of collisions in cryptographic hash function used in digital forensic tools. *Procedia Comput. Sci.* **2017**, *116*, 382–392. [CrossRef]
14. Korus, P. Digital image integrity—A survey of protection and verification techniques. *Digit. Signal Process.* **2017**, *71*, 1–26. [CrossRef]
15. Yunianto, E.; Prayudi, Y.; Sugiantoro, B. B-DEC: Digital evidence cabinet based on blockchain for evidence management. *Int. J. Comput. Appl.* **2019**, *181*, 22–29.
16. Cosic, J.; Cosic, Z. Chain of custody and life cycle of digital evidence. *Comput. Technol. Appl.* **2012**, *3*, 126–129.
17. Giova, G. Improving chain of custody in forensic investigation of electronic digital systems. *Int. J. Comput. Sci. Netw. Secur.* **2011**, *11*, 1–9.
18. Gayed, T.; Lounis, H.; Bari, M. Cyber forensics: Representing and (im) proving the chain of custody using the semantic web. In Proceedings of the IEEE International Conference on Advanced Cognitive Technologies and Applications, Nice, France, 25–29 October 2012; pp. 19–23.
19. Widatama, K.; Prayudi, Y.; Sugiantoro, B. Application of RC4 cryptography method to support xml security on digital chain of custody data storage. *Int. J. Cyber-Secur. Digit. Forensics* **2018**, *7*, 230–237. [CrossRef]
20. 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 IEEE International Conference on Network Softwarization, Paris, France, 24–28 June 2019; pp. 110–114.
21. Bonomi, S.; Casini, M.; Ciccotell, C. B-CoC: A blockchain-based chain of custody for evidences management in digital forensics. *arXiv* **2018**, arXiv:1807.10359, 1–18.
22. Baqir, M.; Saleem, S.; Zulqarnain, R. Protecting digital evidence integrity and preserving chain of custody. *J. Digit. Forensics Secur. Law* **2017**, *12*, 121–130.
23. Sarantinos, N.; Benzaıd, C.; Arabiat, O.; Al-Nemrat, A. Forensic malware analysis: The value of fuzzy hashing algorithms in identifying similarities. In Proceedings of the IEEE International Conference Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 1783–1787.
24. Dodson, J.; Siraj, A. Applying fuzzy hashing to steganography. *Int. J. Future Comput. Commun.* **2015**, *4*, 421–425. [CrossRef]
25. Vikram, S.; Breitinger, F.; Baggili, I. Bytewise approximate matching: The good, the bad, and the unknown. *J. Digit. Forensics Secur. Law* **2016**, *11*, 59–78.
26. Martinez, V.; Álvarez, F.; Encinas, L. State of the art in similarity preserving hashing functions. In Proceedings of the IEEE International Conference on Security and Management, Washington, DC, USA, 12–14 May 2014; pp. 1–7.

27. Lillis, D.; Breitinger, F.; Scanlon, M. Expediting MRSH-v2 approximate matching with hierarchical bloom filter trees. In Proceedings of the IEEE International Conference on Digital Forensics and Cyber Crime, Cham, Switzerland, 9–11 October 2017; pp. 144–157.
28. Breitinger, F.; Baier, H. A fuzzy hashing approach based on random sequences and hamming distance. In Proceedings of the IEEE International Conference on Digital Forensics, Security and Law, Darmstadt, Germany, 2–5 December 2012; pp. 89–100.
29. Naik, N.; Jenkins, P.; Savage, N.; Yang, L. Cyberthreat Hunting—Part 2: Tracking ransomware threat actors using Fuzzy Hashing and fuzzy c-means clustering. In Proceedings of the IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, 23–26 June 2019; pp. 1–6.
30. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Boongoen, T. Fuzzy-Import Hashing: A malware analysis approach. In Proceedings of the IEEE International Conference on Fuzzy Systems, Glasgow, UK, 19–24 July 2020; pp. 1–8.
31. Lu, H.; Zhang, M.; Xu, X.; Li, Y.; Tao Shen, H. Deep fuzzy hashing network for efficient image retrieval. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1–11. [CrossRef]
32. Naik, N.; Jenkins, P.; Savage, N. A Ransomware detection method using fuzzy hashing for mitigating the risk of occlusion of information systems. In Proceedings of the IEEE International Conference on Systems Engineering, Edinburgh, UK, 1–3 October 2019; pp. 1–6.
33. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Asagar, K. Fuzzy hashing aided enhanced yara rules for malware triaging. In Proceedings of the IEEE International Conference on Computational Intelligence, Canberra, ACT, Australia, 1–4 December 2020; pp. 1138–1145.
34. Naik, N.; Jenkins, P.; Gillett, J.; Mouratidis, H.; Naik, K. Lockout-tagout ransomware: A detection method for ransomware using fuzzy hashing and clustering. In Proceedings of the IEEE International Conference on Computational Intelligence, Xiamen, China, 6–9 December 2019; pp. 2792–2796.
35. Wutthikarn, R.; Guang, Y. Prototype of blockchain in dental care service application based on hyperledger composer in hyperledger fabric framework. In Proceedings of the IEEE International Conference on Computer Science and Engineering, Chiang Mai, Thailand, 21–24 November 2018; pp. 1–4.
36. Demichev, A.; Kryukov, A.; Prikhodko, N. The approach to managing provenance metadata and data access rights in distributed storage using the hyperledger blockchain platform. In Proceedings of the IEEE International Conference on Ivannikov Ispras Open Conference, Moscow, Russia, 22–23 November 2018; pp. 131–136.
37. Choi, W.; Won-Ki Hong, J. Performance Evaluation of Ethereum private and test net networks using hyperledger caliper. In Proceedings of the IEEE International Conference on Asia-Pacific Network Operations and Management Symposium, Tainan, Taiwan, 8–10 September 2021; pp. 325–329.
38. Ampel, B.; Patton, M.; Chen, H. Performance modeling of hyperledger sawtooth blockchain. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics, Shenzhen, China, 1–3 July 2019; pp. 59–61.
39. Sukhwani, H.; Wang, N.; Trivedi, K.; Rindos, A. Performance modeling of hyperledger fabric (permissioned blockchain network). In Proceedings of the IEEE International Conference on Network Computing and Applications, Cambridge, MA, USA, 1–3 November 2018; pp. 1–10.
40. Park, I.; Lee, T.; Jang, J. Trade-off in implementation of p2p energy trading over hyperledger blockchain. In Proceedings of the IEEE International Conference on Consumer Electronics—Asia, Seoul, Korea, 1–3 November 2020; pp. 1–4.
41. Frankowski, A.; Dębski, A. Recovery of forensic traces with use of state-of-the-art imaging techniques—system for marking, tracing and maintaining chain of custody. *Issues Forensic Sci.* **2018**, *299*, 52–56. [CrossRef]
42. 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*; Springer: Cham, Switzerland, 2019; pp. 149–168.
43. Burri, X.; Casey, E.; Bolle, T.; Jaquet-Chiffelle, D. Chronological independently verifiable electronic chain of custody ledger using blockchain technology. *Forensic Sci. Int. Digit. Investig.* **2020**, *33*, 300976. [CrossRef]
44. Tanner, A.; Bruno, J. Timely: A Chain of Custody Data Visualizer. In Proceedings of the IEEE Southeast Conference, Huntsville, AL, USA, 11–14 April 2019; pp. 1–5.
45. Dasaklis, T.; Casino, F.; Patsakis, C. Sok: Blockchain solutions for forensics. In *Technology Development for Security Practitioners*; Springer: Cham, Switzerland, 2021; pp. 21–40.
46. Chopade, M.; Khan, S.; Shaikh, U.; Pawar, R. Digital forensics: Maintaining chain of custody using blockchain. In Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 744–747.
47. Fontani, M.; Bianchi, T.; De Rosa, A.; Piva, A.; Barni, M. A forensic tool for investigating image forgeries. *Int. J. Digit. Crime Forensics* **2013**, *5*, 15–33. [CrossRef]