

Received January 14, 2022, accepted January 27, 2022, date of publication January 31, 2022, date of current version February 9, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147809

# Improving Uncertainty in Chain of Custody for Image Forensics Investigation Applications

HANY M. ELGOHARY<sup>1</sup>, SAAD M. DARWISH<sup>2</sup>, AND SALEH MESBAH ELKAFFAS<sup>3</sup>

<sup>1</sup>Expert Counterfeiting and Forgery Research, Forensic Medicine Department, Ministry of Justice, Cairo 68784, Egypt

<sup>2</sup>Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21543, Egypt

<sup>3</sup>College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 1029, Egypt

Corresponding author: Saad M. Darwish (saad.darwish@alexu.edu.eg)

This work was supported in part by the U.S. Department of Commerce under Grant BS123456.


**ABSTRACT** Cybercrime investigations rely heavily on digital evidence to establish links between suspects and the criminal conduct they are allegedly involved in. As a result, digital evidence must be protected since it is complex, volatile, and susceptible to alteration. In the digital evidence method, the chain of custody (CoC) is essential. As a result of the CoC, it is possible to establish that the evidence was never tampered with. Due to the inherent uncertainty of digital evidence, the trustworthiness of the CoC cannot be judged at this time. It is the duty of forensic examiners to challenge this inclination and publicly admit the inherent ambiguity in whatever evidence they use to make their decisions. This article suggests a new paradigm for maintaining the integrity of digital evidence in order to overcome these challenges. To handle the uncertainty generated by error-prone technologies while dealing with CoC documents, the new paradigm used a fuzzy hash inside the blockchain data structure. Traditional hashing methods are only able to tell whether two inputs are precisely the same or not because they are sensitive to even the smallest input changes. Using fuzzy hash functions, we can figure out how dissimilar two images are by comparing their similarities. As an example of how this paradigm may be applied to computer systems and make digital investigations more successful, we utilize image forensics as the focus of an in-depth look at how it works.

**INDEX TERMS** Blockchain, chain of custody, digital evidence, fuzzy hash, image forensic.

## I. INTRODUCTION

Multimedia forensics includes a set of scientific techniques recently proposed for the analysis of multimedia signals (audio, videos, and images) in order to recover probative evidence from them; in particular, such technologies aim to reveal the history of digital contents by (1) identifying the acquisition device that produced the data, (2) validating the integrity of the contents, and (3) retrieving information from multimedia signals [1], [2]. The usual methodology is based on the idea that inherent traces (like digital fingerprints) remain in digital content both during the creation process and any other successive processing; hence, by extracting some digital fingerprints from the data and analyzing their properties, it is possible to have some knowledge of the life cycle of the data [3], [4].

A Chain of Custody (CoC) is a critical process in the management of evidence and investigations. CoC is a term

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo .

that refers to the process of preserving and documenting the chronological history of digital evidence [5]–[10]. 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 [11]–[19].

The scientific problem with the existing chain of custody is that it is impossible to prove that evidence has not been altered with malicious intent through all phases. Several challenges are facing the process of CoC, such as data integrity and the security of CoC documentation. Digital evidence is complex, diffuse, volatile, and easy to change. There are many indications that may be used to identify problems with the management of CoC [10], [20]–[23]: (1) threatens the integrity of digital evidence throughout its lifetime.

(2) Billions of linked devices generate massive amounts of data that must be stored, posing significant challenges 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 software in the computer and information technology fields continues to increase, cybercrime has difficulties in terms of the amount of evidence being examined. (5) The CoC documentation is secure. This is a critical problem since digital evidence may be copied and transferred to other systems. (6) CoC adaptability and capacity: this issue 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, as well as providing an auditing facility to ensure the accuracy of forensic tools and their application procedures. Furthermore, it must preserve the artefacts of the evidence for digital evidence to be admissible in court [10], [19]. 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 [24]. Blockchain technology is also a potential method for evidence verification and management in the area of digital forensics, and it is being extensively explored [14].

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 [8]. Numerous methods are discussed in detail in order to identify image forgery. They are categorized as active or passive algorithms [9]. 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 of 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.

Every piece of digital data (evidence in our case) 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. Accordingly, merging fuzzy hash within a blockchain to preserve the chain of custody was investigated in this paper for the first time in the field of image forensics investigation. In the suggested framework, investigators can cope with allowed digital evidence manipulation using fuzzy

hash algorithms, which are unsuccessful when employing standard hash.

The objective of this study that proves CoC is to demonstrate that the evidence has not been tampered with at any point throughout the investigation. In order to handle this problem, this paper sets out a new framework for the integrity of digital evidence and CoC documents. More precisely, the proposed framework focuses on fuzzy hashing inside blockchain technology, different from traditional cryptographic hash algorithms such as MD5 or SHA-256, which are designed to be sensitive to small input modifications and can only determine if the inputs are exactly the same or not. Fuzzy hash functions hold a certain tolerance for changes and can tell how different two images are by comparing their similarity.

This paper focuses on the research of protecting digital evidence with the uncertainty that it is still a challenging research topic and relatively less touched by researchers. Traditional blockchain-based chains of custody are 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 may 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 of digital evidence, while using conventional hash methods is ineffective in this situation.

The remainder of this paper will be structured as follows. Section II discusses several similar works and their benefits and drawbacks. The suggested model is described in Section III. Section IV outlines the experiments used to verify the proposed model, and Section V concludes the paper.

## II. 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 [25] suggested a Blockchain-based Chain of Custody (B-CoC) to dematerialize the CoC procedure while ensuring the integrity of gathered evidence and owner traceability. The B-CoC was shown to be an effective aid for the CoC process during the performance assessment. However, the degree of anonymity for validators must be increased without modifying security attributes. Similarly, the authors of [19] used Blockchain to integrate the Digital Evidence Cabinet (DEC) architecture. This prototype is referred to as the (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. That it needs to significantly strengthen the protection of digital evidence, such as through the use of encryption.

The work in [12] established a reliable time stamping technique for protecting digital evidence during the investigative process. The timestamp will be acquired from the 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 [16], in which the authors utilized a variety of security techniques to protect the integrity of the digital evidence, including (CRC-Hash Functions-Digital Signatures). SHA-512 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 [23] encrypted the XML structure of 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 [26] evaluated two automated disc 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 integrity of the evidence.

Tian et al. [14] 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 that ensures the authenticity of evidence and strikes an appropriate balance between privacy and traceability. However, adding a new node to the blockchain takes an inordinate amount of time to download and validate the blockchain.

The primary difference between the proposed model and the previous blockchain-based image forensics frameworks is that the proposed model analyzes the blockchain validity (evidence items) using fuzzy hashing rather than traditional hashing in order to extend the ability of related work to deal with evidence item modifications caused by benign or malicious attacks. When the resemblance between two blocks exceeds 95%, the block is recognized as original evidence.

### III. 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). In respect to the CoC, the blockchain technology,

especially when combined with fuzzy hashing, has the potential to provide a tamper-proof recording of evidence. The suggested model’s fundamental process is shown in Fig.1.

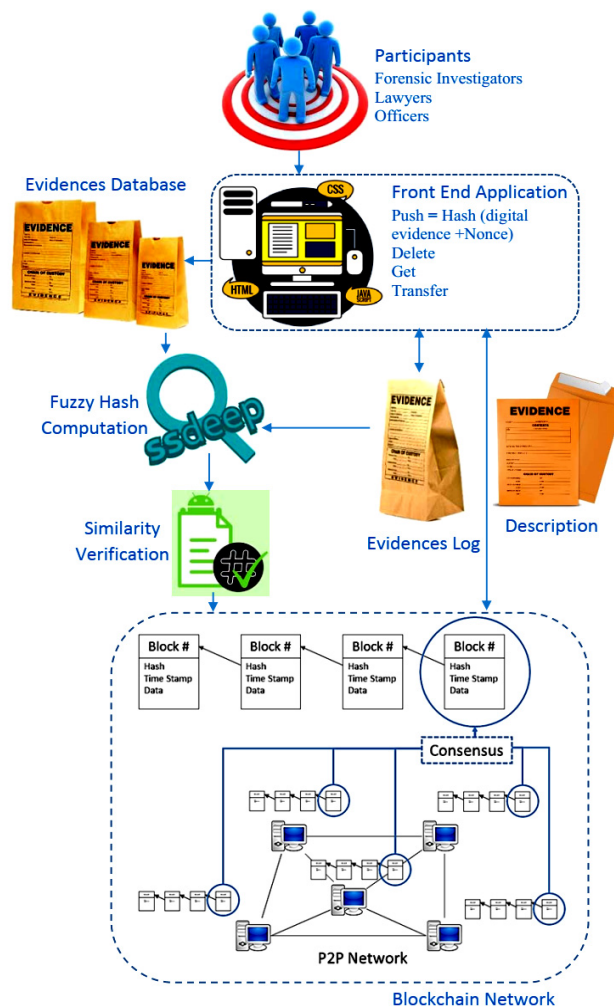


FIGURE 1. Proposed model for protecting digital evidence integrity under uncertainty.

In the proposed model, the philosophy of using fuzzy hashing within the blockchain in the proposed framework is based on the fact that 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. Permitted modification of digital evidence may be effectively addressed by using fuzzy hash functions, while traditional hash techniques are useless in this scenario. By comparing blocks to all nodes in the blockchain network using fuzzy hash similarity, the digital forensics investigator will be able to verify their authenticity. Fuzzy hashing can locate similar images and match altered images. Furthermore, it accounts for the uncertainty associated with

evidence item changes. Each framework's stage will be discussed in depth in the following subsections.

### A. PARTICIPANTS

Authorized parties (forensic investigation) gather digital evidence (images) and then register them in the blockchain. The lawyer, police, defense, and court all participate in the forensic investigation because they need information regarding the CoC at various points throughout the investigation. Only authorized parties have access to the data associated with a specific piece of evidence [10]. 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 [25].

### B. FRONT END

This part is intended to serve as an interface for authorization, access permissions, and 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 [19], [24]. 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 [25].

### C. THE 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, 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 [19]: (1) Validator nodes are primarily in charge of keeping a copy of the blockchain, validating transactions, and creating, proposing, and adding blocks to the chain (i.e., participating 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. The Evidence Log runs a smart contract which exposes four primitives [25]:

- Create Evidence (ID, description): stores a new evidence entry in the blockchain with the specified ID and description, setting the submitter identity as the creator and current owner of the evidence.
- Transfer (ID, new owner): transfers the ownership of an evidence (registering the handover). It fails if the issuer is not the current owner.
- Remove Evidence (ID): removes an evidence entry. It fails if the issuer is not the creator.
- Get Evidence (ID): returns the information in the evidence entry. Namely, the ID, description, creator and

all owners with the time of each change of ownership. Herein, every users in the network can query the Evidence Log to get the entry of evidence (which contains all relevant information except the evidence itself).

### D. BLOCKCHAIN

A blockchain is a decentralized ledger that is maintained by trustless nodes in a peer-to-peer network. Data is stored on the blockchain in blocks that are linked together through a connection to the hash value of each block. It is not feasible to modify data in the midst of a block [19]. The first responder initiates the forensic-chain by hashing digital evidence (image) and securely storing it on the blockchain through the smart contract. Additional information like 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 is 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 [13].

The presented framework is based on a private and permissioned blockchain, although permissioned private is a complex network. This choice has been driven by the authentication requirement of the CoC process, which does not allow unauthorized and untrusted parties to manage digital evidence and thus be in the network. Permissioned blockchain networks are used by organizations that need to more tightly control and protect their blockchain. In this case, only authorized users are maintaining the blockchain, and the data may only be available to those within the blockchain network.

The rationale for choosing blockchain as the core network even though some newer hyperledger networks are available lies in the difference between blockchain and hyperledger networks. The most significant point to remember is that a blockchain is simply one type of distributed ledger. Although a blockchain is a series of blocks, distributed ledgers do not need such a chain. Furthermore, distributed ledgers do not require proof of work and provide (potentially) superior scalability possibilities. Unlike the blockchain, a distributed ledger does not necessarily need to contain a data structure in blocks. A blockchain is an extended series of blocks that are connected via encryption, and each block provides a digital record of a batch of certified evidence. From the above, it becomes clear to us that the nature of the current problem needs the block's data structure to store different data for each validated evidence and proof of work mechanism authentication.

#### 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 [27].

Rolling hashing generates “segments” of conventional hash strings by generating a pseudo-random value depending on the context of the input. In contrast, PH (Piecewise Hashes), like traditional hashes, generates a final checksum for the entire image. 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 [27], [28].

## 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 [29]. There are eight well-known approximation matching algorithms, including the following: (ssdeep, sdhash, mrsh-v2, bbHash, mvHash-B, SimHash, saHash, 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 [29]. While ssdeep is the most well-known CTPH method used 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 [30]. Breitingner published this method in 2013 as a mix of ssdeep and sdhash.

The ssdeep algorithm [30] computes the similarity of two files based on their signatures throughout the comparison process. 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 [27]. 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 [27].

Mrsh-v2 overcomes ssdeep’s limitations and becomes quicker than sdhash [29]. 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 (a 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 [31].

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 in it. If a node has a certain number of consecutive hashes, this is considered a match [31].

## 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 [32]. 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 [29]. Byte-wise has a number of restrictions, including [29]. (1) It is unable to detect similarities at a higher level of abstraction, for example, semantically. As a result, two image files that contain the same semantic image but are stored in various file kinds and formats are unable to be properly matched. (2) Due to the absence of a universally accepted definition of similarity, not all types of byte-level similarity are equally useful, since certain artefacts (e.g., headers and footers) are trivial and result in false positives.

This research focuses on the second type, 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 the same, but the images will be identical [29]. A comparison function is required to compare two hash values. 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 identical, 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.

### E. PEER TO PEER NETWORK

A Peer-to-Peer (P2P) network 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 the 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 [14].

#### 1) CONSENSUS MECHANISM

The blockchain consensus process selects a node to generate and broadcast the next block of the blockchain and ensures that each node's blockchain is consistent [14]. 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 [19].

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 incentivise such a system, solving a block also leads to the mining of a certain number of bitcoins, which serves as an incentive for block makers (often referred to as miners) [25]. 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 [25]. 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 to a loss of personal reputation and, eventually, expulsion from the validator set [25].

### F. 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 [33]. 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 from anybody [10]. 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 [10].

The proposed model in this article is based on HLF and HLC, which offers the following major benefits. [10], [34] (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. (5) Ethereum is probably not the ideal cryptocurrency to use for crime scene investigation. Digital forensic investigations need confidentiality and are conducted by genuine and trustworthy parties.

From a functional standpoint, the HLF network's nodes are classified as follows [34]: (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, as well as 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 & HLC) platform, a variety of concepts are employed; the most important of which are assets, participants, and network-stored transactions. (1) Assets: an asset is anything of value that can be traded or shared over a network.

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 model file. It is possible to generate new instances of the modelled 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 & HLC) platform. (3) Transactions are used to explain the activities that participants may undertake on assets as they travel through the network. Transactions in the proposed model either record information about the evidence or the evidence transfer event on the network.

**G. EVIDENCE DATABASE**

The Evidences DB 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, and officers). Additionally, 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 Evidence database). To begin with, 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 with 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 only keep 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 [25]. See [35]–[41] for more information about protecting digital evidence integrity and preserving the chain of custody.

**IV. 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 produces reports that include performance metrics like 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 of DDR3 RAM laptop, with the

Windows 10 operating system. The code was written in Python using Python 3.6 software.

**A. 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 model’s transactional efficiency, we created a test file that targeted two primary functionalities of our model, 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.

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

Round	Send Rate (tps)	Max Latency (sec)	Min Latency (sec)	Avg Latency (sec)	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 model.**

Round	Send Rate (tps)	Max Latency (Sec)	Min Latency (Sec)	Avg Latency (Sec)	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 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 throughput obtained in 2-organization-1-peer and 3-organization-1-peer network architectures is 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 characteristics of Hyperledger-based consortium blockchains.

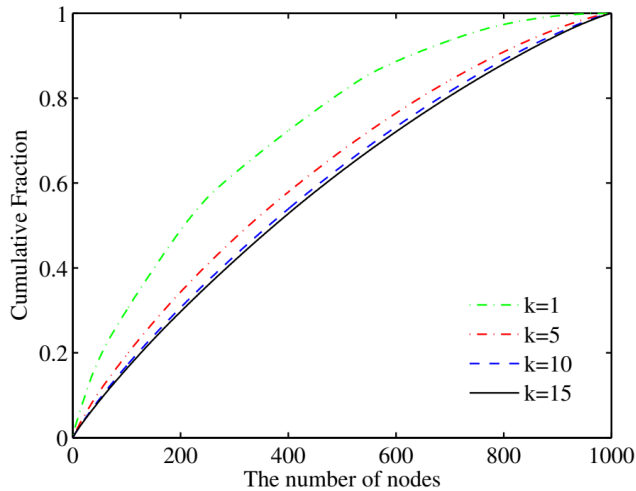


FIGURE 2. The cumulative distribution of generated block.

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 that 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  $10^5$  blocks sequentially, counting the blocks generated by each node. The cumulative percentage of produced blocks containing  $x$  nodes is shown in Fig. 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.

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 is set to one and the group size variable,  $h$ , is set to three bits for the topology. A block may contain no more than 2000 transactions. Following that, we determined the blockchain’s storage capacity on each node. We are 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 represents the blockchain’s load balancing mechanism. Three times, we do the experiment. 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.

Fig. 3 illustrates the blockchain’s size as a function of the block count. The maximum, mean, and minimum blockchain

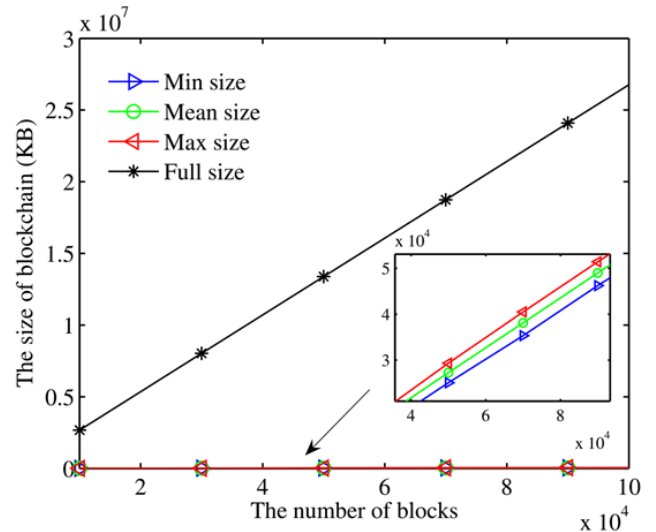


FIGURE 3. The size of blockchain.

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 in size 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 next set of experiments to determine the time required to conduct a full search, and to determine the approach’s success in locating the 100 “illegal” files included verbatim in the hard disc image, as well as the 40 files from the image that are similar to “illegal” files, as defined by MRS<sub>H</sub>-v2. A collection of simulated “known-illegal” images consisting of 4,000 images plus 140 more images as follows: Within the 4,000 “illegal” images, there are 100 images. As determined by MRS<sub>H</sub>-v2, 40 images that are not included in the “illegal” images but show a high degree of resemblance to images in the corpus, as determined by MRS<sub>H</sub>-v2.

The main measure is the time needed to execute the whole process, which includes the time required to construct the tree, search the tree, and perform pairwise comparisons on the leaves. MRS<sub>H</sub>-v2 ran for a total of 2,592 seconds. Fig. 4 illustrates the running times. The tree was constructed using the smaller sample of 4,000 “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 for the tree and the time spent doing leaf comparisons. As anticipated, the more leaf nodes resulted in the quickest execution time. The race lasted 1,182 seconds (a 54 percent reduction in 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.

In this set of experiments, fuzzy hashes based on MRS<sub>H</sub>-v2 and conventional hashes were compared in terms of response time while establishing the root hash for the



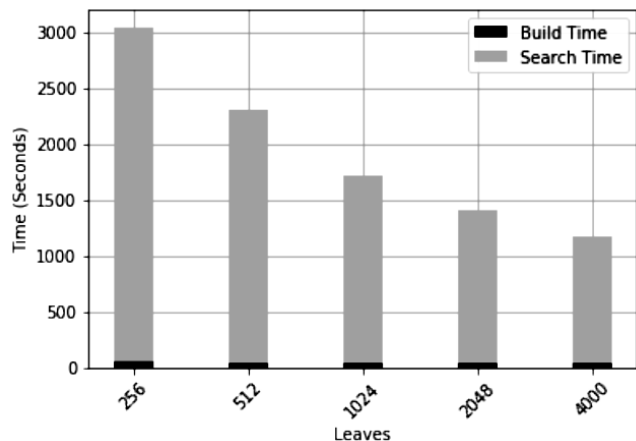


FIGURE 4. Time to search for planted evidence (including pairwise comparisons).

TABLE 3. Response time for the mined block using SHA256 vs fuzzy hash (seconds).

	No. of Blocks						
	100	300	400	500	600	800	1000
Fuzzy Hash based on MRSH-v2	0.3	0.9	1.1	2	2.4	4	6
Traditional Hash based on SHA256	0.4	1.25	1.55	3	3.3	5.2	7.8

newly-constructed bloom filter. Responding to a transaction, mining a new block, and producing a text file containing the block information constitutes the node’s response time. You can see in Table 3 how long it takes to respond to a certain number of minded blocks. The results show that as the number of mined blocks grows, so does the response time.

An average of 30% faster response time was achieved by using a fuzzy hash rather than a regular hash. A real-time digital inquiry application based on this approach is now possible. There is a possibility that fuzzy hash uses the MD5 method to build a bloom filter, which is equivalent in complexity to SHA256, but takes less time to perform than SHA256 does. In comparison to SHA256, MD5 generates hashes with a smaller key size.

**B. SECURITY ANALYSIS**

As far as forensics is concerned, both blacklisting and whitelisting attacks are discussed in this section. From the perspective of an attacker, anti-blacklisting and anti-whitelisting may be used to conceal information. 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 can’t 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 [42], in the worst case, each building block has a hamming distance and a “one-bit-change” is enough to manipulate the triggering. In this case, an active adversary needs to change one bit for each position. Actually, a lot more than 100 more changes need to be made as there are also positions where the hammering distance has a 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 the 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 pre image-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.

**V. CONCLUSION**

Cybercrime may be exposed through a variety of digital forensics operations. The integrity and credibility of the digital evidence in a single process for managing the chain of custody are critical components of these operations. This article provides an overview of the extent to which the digital chain of custody issue faces problems and difficulties, particularly the issue of uncertainty, as well as the breadth of research that may be conducted to contribute to the issue of the digital chain of custody. 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. Additionally, the main aim of this study was to determine the viability of fuzzy hashing for assessing similarities.

We developed and tested a prototype of a forensic chain model based on 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. A 54.0% reduction in the time required for an all-against-all pairwise comparison was achieved. Furthermore, an average of 30% faster response time was achieved by using a fuzzy hash rather than a regular hash.

Further research is planned in light of the encouraging results of the experiments reported in this article.

Currently, files are assigned to leaf nodes in a round-robin manner during tree construction. It is conceivable that a more efficient allocation method might be employed for trees with many files represented at each leaf (e.g., to allocate similar files to the same leaf node). Additionally, the present model employs balanced trees, ensuring that all successful searches descend to the same depth in the tree. In certain cases, an imbalanced tree may be desirable in order to speed up some of the most frequent queries.

## REFERENCES

- [1] A. A. Khan, M. Uddin, A. A. Shaikh, A. A. Laghari, and A. E. Rajput, "MF-ledger: Blockchain hyperledger sawtooth-enabled novel and secure multimedia chain of custody forensic investigation architecture," *IEEE Access*, vol. 9, pp. 103637–103650, 2021.
- [2] I. Amerini, A. Anagnostopoulos, L. Maiano, and L. Celsi, "Deep learning for multimedia forensics. Foundations and Trends," *Comput. Graph. Vis.*, vol. 12, no. 4, pp. 309–457, 2021.
- [3] S. Battiato, O. Giudice, and A. Paratore, "Multimedia forensics: Discovering the history of multimedia contents," in *Proc. 17th Int. Conf. Comput. Syst. Technol.*, Jun. 2016, pp. 5–16.
- [4] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," *Electron. Imag.*, vol. 2017, no. 7, pp. 77–86, Jan. 2017.
- [5] S. Dosis, I. Homem, and O. Popov, "Semantic representation and integration of digital evidence," *Proc. Comput. Sci.*, vol. 22, pp. 1266–1275, Oct. 2013.
- [6] Y. Prayudi and A. Sn, "Digital chain of custody: State of the art," *Int. J. Comput. Appl.*, vol. 114, no. 5, pp. 1–9, Mar. 2015.
- [7] N. Campbell, T. Goodyear, W. Messer, E. Stuart, and J. Fairbanks, "Digital witness: Remote method for volunteering digital evidence on mobile devices," in *Proc. IEEE Int. Symp. Technol. for Homeland Secur. (HST)*, Woburn, MA, USA, Oct. 2018, pp. 1–5.
- [8] J. Patel and N. Bhatt, "Review of digital image forgery detection," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 7, pp. 152–155, Jul. 2017.
- [9] A. Varkey and L. Nair, "Robust image forgery detection and classification in copy-move using SVM," *Int. J. Adv. Res. Trends Eng. Technol.*, vol. 5, no. 12, pp. 89–93, Apr. 2018.
- [10] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in hyperledger composer," *Digit. Invest.*, vol. 28, pp. 44–55, Jan. 2019.
- [11] M. Sadiku, A. Shadare, and S. Musa, "Digital chain of custody," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 7, pp. 117–118, Jul. 2017.
- [12] J. Čosić and M. Baca, "(Im) Proving chain of custody and digital evidence integrity with time stamp," in *Proc. IEEE Int. Conf. Conv., Opatija*, Croatia, Jun. 2010, pp. 1226–1230.
- [13] L. Auqib and M. Roohie, "Forensic-chain: Ethereum blockchain based digital forensics chain of custody," *Sci. Practical Cyber Secur. J.*, vol. 1, no. 2, pp. 21–27, Dec. 2017.
- [14] Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, "Block-DEF: A secure digital evidence framework using blockchain," *Inf. Sci.*, vol. 491, pp. 151–165, Apr. 2019.
- [15] J. Čosić, Z. Čosić, and M. Baca, "An ontological approach to study and manage digital chain of custody of digital evidence," *Int. J. Inf. Organizational Sci.*, vol. 35, no. 1, pp. 1–13, Jun. 2011.
- [16] S. Saleem, O. Popov, and R. Dahman, "Evaluation of security methods for ensuring the integrity of digital evidence," in *Proc. Int. Conf. Innov. Inf. Technol.*, Apr. 2011, pp. 220–225.
- [17] Z. Rasjid, B. Soewito, G. Witjaksono, and A. Edi, "A review of collisions in cryptographic hash function used in digital forensic tools," *Proc. Comput. Sci.*, vol. 116, pp. 382–392, Oct. 2017.
- [18] P. Korus, "Digital image integrity—A survey of protection and verification techniques," *Digit. Signal Process.*, vol. 71, pp. 1–26, Aug. 2017.
- [19] E. Yunianto, Y. Prayudi, and B. Sugiantoro, "B-DEC: Digital evidence cabinet based on blockchain for evidence management," *Int. J. Comput. Appl.*, vol. 181, no. 45, pp. 22–29, Mar. 2019.
- [20] J. Cosic and Z. Cosic, "Chain of custody and life cycle of digital evidence," *Comput. Technol. Appl.*, vol. 3, no. 2, pp. 126–129, Feb. 2012.
- [21] G. Giova, "Improving chain of custody in forensic investigation of electronic digital systems," *Int. J. Comput. Sci. Netw. Secur.*, vol. 11, no. 1, pp. 1–9, Jan. 2011.
- [22] T. Gayed, H. Lounis, and M. Bari, "Cyber forensics: Representing and (Im) proving the chain of custody using the semantic web," in *Proc. IEEE Int. Conf. Adv. Cognit. Technol. Appl.*, Paris, France, 2012, pp. 19–23.
- [23] K. Widatama, Y. Prayudi, and B. Sugiantoro, "Application of RC4 cryptography method to support XML security on digital chain of custody data storage," *Int. J. Cyber-Secur. Digit. Forensics*, vol. 7, no. 3, pp. 230–237, 2018.
- [24] S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiaeles, D. Kavallieros, E. Bellini, and C. Pavue, "Blockchain solutions for forensic evidence preservation in IoT environments," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Paris, France, Jun. 2019, pp. 110–114.
- [25] S. Bonomi, M. Casini, and C. Ciccotelli, "B-CoC: A blockchain-based chain of custody for evidences management in digital forensics," 2018, *arXiv:1807.10359*.
- [26] M. Shah, S. Saleem, and R. Zulqarnain, "Protecting digital evidence integrity and preserving chain of custody," *J. Digit. Forensics, Secur. Law*, vol. 12, no. 2, pp. 121–130, Jun. 2017.
- [27] N. Sarantinos, C. Benzaïd, O. Arabiat, and A. Al-Nemrat, "Forensic malware analysis: The value of fuzzy hashing algorithms in identifying similarities," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1783–1787.
- [28] J. Dodson and A. Siraj, "Applying fuzzy hashing to steganography," *Int. J. Future Comput. Commun.*, vol. 4, no. 6, pp. 421–425, 2015.
- [29] V. Harichandran, F. Breiting, and I. Baggili, "Byte-wise approximate matching: The good, the bad, and the unknown," *J. Digit. Forensics, Secur. Law*, vol. 11, no. 2, pp. 59–78, 2016.
- [30] V. Martinez, F. H. Álvarez, and L. Encinas, "State of the art in similarity preserving hashing functions," in *Proc. IEEE Int. Conf. Secur. Manage., Las Vegas, NV, USA*, Mar. 2014, pp. 1–7.
- [31] D. Lillis, F. Breiting, and M. Scanlon, "Expediting MRS-H-v2 approximate matching with hierarchical Bloom filter trees," in *Proc. IEEE Int. Conf. Digit. Forensics Cyber Crime*, Cham, Switzerland, Oct. 2017, pp. 144–157.
- [32] F. Breiting and H. Baier, "A fuzzy hashing approach based on random sequences and Hamming distance," in *Proc. IEEE Int. Conf. Digit. Forensics, Secur. Law*, May 2012, pp. 89–100.
- [33] R. Wutthikarn and Y. G. Hui, "Prototype of blockchain in dental care service application based on hyperledger composer in hyperledger fabric framework," in *Proc. 22nd Int. Comput. Sci. Eng. Conf. (ICSEC)*, Chiang Mai, Thailand, Nov. 2018, pp. 1–4.
- [34] A. Demichev, A. Kryukov, and N. Prikhodko, "The approach to managing provenance metadata and data access rights in distributed storage using the hyperledger blockchain platform," in *Proc. Ivannikov Ispras Open Conf. (ISPRAS)*, Moscow, Russia, Nov. 2018, pp. 131–136.
- [35] C. A. Frankowski and M. A. Dębski, "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.*, vol. 299, no. 1, pp. 52–56, 2018.
- [36] H. Al-Khateeb, G. Epiphaniou, and H. Daly, "Blockchain for modern digital forensics: The chain-of-custody as a distributed ledger," in *Blockchain and Clinical Trial*. Cham, Switzerland: Springer, 2019, pp. 149–168.
- [37] X. Burri, E. Casey, T. Bolle, and D. Jaquet-Chiffelle, "Chronological independently verifiable electronic chain of custody ledger using blockchain technology," *Forensic Sci. Int., Digit. Investigation*, vol. 33, 300976, pp. 1–11, 2020.
- [38] A. Tanner and J. Bruno, "Timely: A chain of custody data visualizer," in *Proc. IEEE Southeast Conf.*, Apr. 2019, pp. 1–5.
- [39] T. Dasaklis, F. Casino, and C. Patsakis, "SOK: Blockchain solutions for forensics," in *Technology Development for Security Practitioners*. Cham, Switzerland: Springer, 2021, pp. 21–40.
- [40] M. Shah, S. Saleem, and R. Zulqarnain, "Protecting digital evidence integrity and preserving chain of custody," *J. Digit. Forensics, Secur. Law*, vol. 12, no. 2, pp. 122–129, 2017.
- [41] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A forensic tool for investigating image forgeries," *Int. J. Digit. Crime Forensics*, vol. 5, no. 4, pp. 15–33, Oct. 2013.
- [42] N. Naik, P. Jenkins, N. Savage, L. Yang, K. Naik, J. Song, T. Boongoen, and N. Iam-On, "Fuzzy hashing aided enhanced YARA rules for malware triaging," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Canberra, ACT, Australia, Dec. 2020, pp. 1138–1145.



**HANY M. ELGOHARY** received the B.Sc. degree in special chemistry from the Faculty of Science, Al-Azhar University, Egypt, in 2002, the Diploma degree in analytical biochemistry from the Faculty of Science, Alexandria University, Egypt, in 2004, the M.Sc. degree in information technology from the Department of Information Technology, Institute of Graduate Studies and Research (IGSR), Alexandria University, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Information Technology, Institute of Graduate Studies and Research, for a thesis in forensic Science. Since 2007, he has been an Expert in counterfeiting and forgery research with the Forensic Medicine Department, Ministry of Justice, Alexandria, Egypt. His research interests include machine learning, pattern recognition, image processing, and digital forensics.



**SAAD M. DARWISH** received the B.Sc. degree in statistics and computer science from the Faculty of Science, Alexandria University, Egypt, in 1995, the M.Sc. degree in information technology from the Department of Information Technology, Institute of Graduate Studies and Research (IGSR), University of Alexandria, in 2002, and the Ph.D. degree from Alexandria University, for a thesis in image mining and image description technologies. Since June 2017, he has been a Professor with

the Department of Information Technology, IGSR. He is the author or coauthor of more than 50 papers publications in prestigious journals and top international conferences and also received several citations. He has served as a reviewer for several international journals and conferences. He has supervised around 80 M.Sc. and Ph.D. students. His research interests include image processing, optimization techniques, security technologies, database management, machine learning, biometrics, digital forensics, and bioinformatics.



**SALEH MESBAH ELKAFFAS** received the B.Sc. degree in electronics and telecommunications engineering, the M.Sc. degree in information content in remote sensing, and the Ph.D. degree under the channel system supervision from the Center for Remote Sensing, Imperial College, U.K., and Alexandria University. He is currently an Associate Professor with the Department of Information Systems, College of Computing and Information Technology, Arab Academy for Science,

Technology and Maritime Transport (AAST), where he is also the Director of the Remote Sensing and Spatial Studies Unit, College of Engineering and Technology. He has over 35 years of experience in the field of remote sensing, GIS, computing and information technology. Over the span of his career, he has published over 80 papers and book chapters in the fields of remote sensing, geographic information systems (GIS), digital image processing, advances in intelligent systems and computing, applied geomatics, sensing and imaging, and environmental management. His research interests include satellite remote sensing, digital image processing, geographic information systems (GIS), global positioning systems (GPS), and information technology and environmental management.

...