IF-DSS: A forensic investigation framework for decentralized storage services

By:
Jihun Son, Gyubin Kim, Hyunwoo Jung, Jewan Bang, Jungheum Park

From the proceedings of
The Digital Forensic Research Conference
DFRWS APAC 2023
Oct 17-20, 2023

Forensic Science International: Digital Investigation 46 (2023) 301611

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 APAC - Proceedings of the Third Annual DFRWS APAC

IF-DSS: A forensic investigation framework for decentralized storage services

Jihun Son [a], Gyubin Kim [b], Hyunwoo Jung [c], Jewan Bang [d], Jungheum Park [a,*]

[a] School of Cybersecurity, Korea University, 145 Anam-Ro, Seongbuk-Gu, Seoul, South Korea
[b] AlpineLab, 169-16 Gasan Digital 2-ro, Geumcheon-gu, Seoul, South Korea
[c] AhnLab, 220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea
[d] Cyber Investigation Bureau, National Office of Investigation, Korean National Police Agency, Seoul, South Korea

ARTICLE INFO

Keywords:
digital forensics
Forensic framework
Decentralized storage
IPFS
Filecoin

ABSTRACT

Decentralized storage services are growing in popularity owing to their lower costs, increased resilience, and privacy compared with traditional cloud storage services. However, these characteristics also attract malicious actors, who abuse them to create phishing URLs, distribute malware, infringe on copyrights, and conduct other crime-related activities. Investigating these services is challenging because of their censorship resistance and decentralization, which renders the existing methodologies for cloud-based storage services and peer-to-peer-based file-sharing services insufficient. To address these challenges, we introduce a novel forensic investigation framework that encompasses identifying, collecting, examining, analyzing potential evidence, and preventing the further distribution of the content. The framework works on each node, peer, gateway, and Internet area of the decentralized storage services, integrating investigation steps on both remote and local sides. The usefulness and applicability of the proposed framework were demonstrated through case studies involving phishing and large-scale file sharing using *IPFS* with *Filecoin*.

1. Introduction

As blockchain and non-fungible tokens (NFTs) gained prominence in 2021, decentralized storage services (hereinafter referred to as DSSs) emerged as an alternative method for storing user data outside of the blockchain. DSS is regarded as an innovative data storage solution because of its lower cost and greater resilience than traditional cloud storage services. *Filecoin*, a well-known DSS and cryptocurrency, has a market capitalization of over two billion dollars, ranking 31st in the overall coin market (as of May 14, 2023) (Filecoin Price, 2023).

DSSs are also abused illegally. The privacy and censorship resistance offered by DSS, along with low costs and high resilience, create an attractive option for criminals. Currently, more than 300,000 phishing samples utilizing the InterPlanetary File System (`IPFS`) are being discovered every month, and this number is expected to increase in the future (Kaspersky, 2023). Anna's Archive, part of the Z-Library project, has uploaded approximately six million books to `IPFS`, raising concerns about copyright infringements (Anna's blog, 2022). *BitTorrent* has created an ecosystem that rewards file-sharing nodes with their cryptocurrency, leading to growing concerns about large-scale copyright infringement through BitTorrent File System (`BTFS`). Cisco Talos and Trend Micro discovered various malware samples distributed using *IPFS* (Cisco Talos, 2022; Trend Micro, 2023). As such, DSSs are increasingly being used in illegal activities such as phishing, copyright infringement, and malware distribution.

Investigating a DSS forensically is challenging because of its characteristics, which include censorship resistance and decentralization. DSS providers possess limited user data and are often reluctant to cooperate with law enforcement agencies. Specifically, *IPFS* and *Filecoin* can be utilized without requiring user registration, and their service providers do not retain any uploaded content (IPFS docs; Set up - Filecoin docs). *Storj* has a warrant canary that informs users when a government requests data disclosure (Storj's Canary). Given the minimal assistance from service providers, investigative agencies must be able to collect potential evidence, analyze user activities, and prevent further content distribution by themselves. However, because all data uploaded to the DSS are distributed across the node, peer, gateway, and Internet areas, it is challenging to identify and collect potential digital evidence

* Corresponding author.
E-mail addresses: hunjison@korea.ac.kr (J. Son), kgb@alpinelab.io (G. Kim), hyunwoo.jung@ahnlab.com (H. Jung), jwbang@police.go.kr (J. Bang), jungheumpark@korea.ac.kr (J. Park).

https://doi.org/10.1016/j.fsidi.2023.301611

Available online 13 October 2023

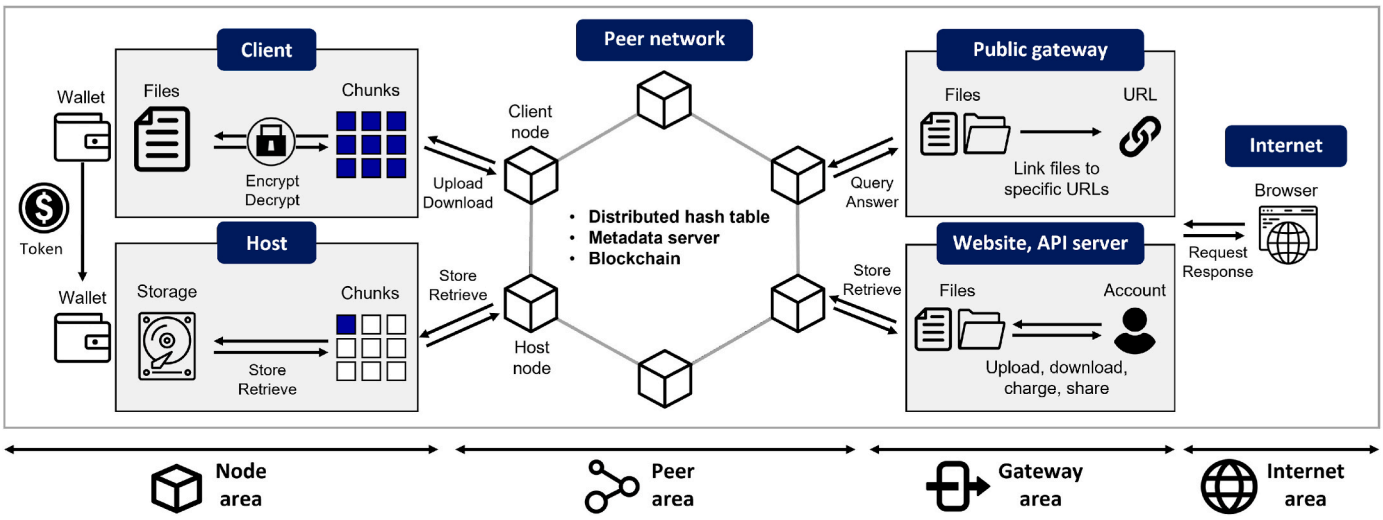

Fig. 1. Four areas that comprise a general ecosystem of decentralized storage services.

using existing digital forensic investigation methodologies.

1.1. Contribution

The main contributions of this study are:

- We identify four areas that comprise a DSS ecosystem and further explain the six main features of a DSS from the perspective of digital forensic investigation.
- We propose *IF-DSS*, a forensic investigation framework for decentralized storage services. In consideration of investigating both on the remote and local sides, our framework involves collecting, examining, and analyzing potential digital evidence from all four areas of a DSS and helps prevent further content distribution.
- We demonstrate the usefulness and applicability of the proposed framework by applying it to two case studies on *IPFS* along with *Filecoin*. We also provide a dataset and a proof-of-concept tool developed during this research.

The remainder of this paper is organized as follows: Section 2 presents the four areas that consist the ecosystem of DSS and describes potential scenarios in which DSS can be abused for criminal activities. Section 3 outlines related works. Section 4 introduces *IF-DSS* and provides each procedure in detail. Section 5 presents two case studies related to *IPFS* with *Filecoin*, demonstrating the usefulness and applicability of our framework. In Section 6, we underscored the unique features of the proposed framework through a comparative study with the existing investigation frameworks for cloud and P2P-based services, and discuss the limitations of the current version of the framework. Finally, Section 7 concludes the paper.

2. Background and potential crime scenarios

2.1. Background: an overview of the DSS

We will explain how DSS operates through four areas, as illustrated in Fig. 1. The four areas simplify the structure of the DSS, making it easier to understand.

2.1.1. Node area

The node area refers to the local storage of client and host nodes. Client nodes upload or download their own files, while host nodes provide their own additional storage. When a client node uploads a file to a DSS, it is divided into chunks and might be encrypted depending on the service. Host nodes store only chunks of the file, and these chunks might be encrypted, making it difficult to identify complete files in the host node's storage. For some DSSs that do not encrypt chunks, there are companies called *pinning service* which guarantee entire chunks of the file stored on them. The client compensates the host through a cryptocurrency for using their additional storage, as stipulated by the smart contract of the DSS.

2.1.2. Peer area

In the peer area, many nodes communicate by exchanging data chunks and their metadata. Metadata refers to the information required to reconstruct chunks to an original file, such as a list of chunks of the file and the storage location. To store and manage metadata, DSS employs one or more methods among distributed hash table (DHT), metadata server, and blockchain. *IPFS* builds a DHT on the peer network using the *Kademlia* algorithm (DHT - IPFS docs). *BitTorrent* and *Storj* operate metadata servers called *trackers* and *satellites* respectively, and anyone can operate these servers (BitTorrent Tracker; Storj's Satellite). *Arweave* stores and manages content and metadata through a blockchain-like structure called *Blockweave* (Williams et al., 2019).

2.1.3. Gateway area

The gateway area comprises of a public gateway, service website, and web API server. A public gateway is a server that generates a public URL, allowing data in the peer area to be accessed via the HTTP protocol on the surface web. Each DSS has an official public gateway, and in some cases, third-party public gateways are actively running, such as *IPFS* (Public gateway checker of IPFS). A service website is used in some DSS for tasks such as signing up, file upload and download, file sharing, payment, and more. An API server provides functions similar to the website, but offers additional functions such as metadata and transaction lookup.

2.1.4. Internet area

The Internet area is where URLs created in the gateway area are shared and where publicly accessible blockchain-related data exists. Malicious actors share URLs for accessing resources on DSS via email, websites, and social media (Trust Wave, 2022; Anna's archive, 2023). In websites known as blockchain explorers, users can browse and explore the data of a blockchain network, which includes the transaction history and metadata of shared content.

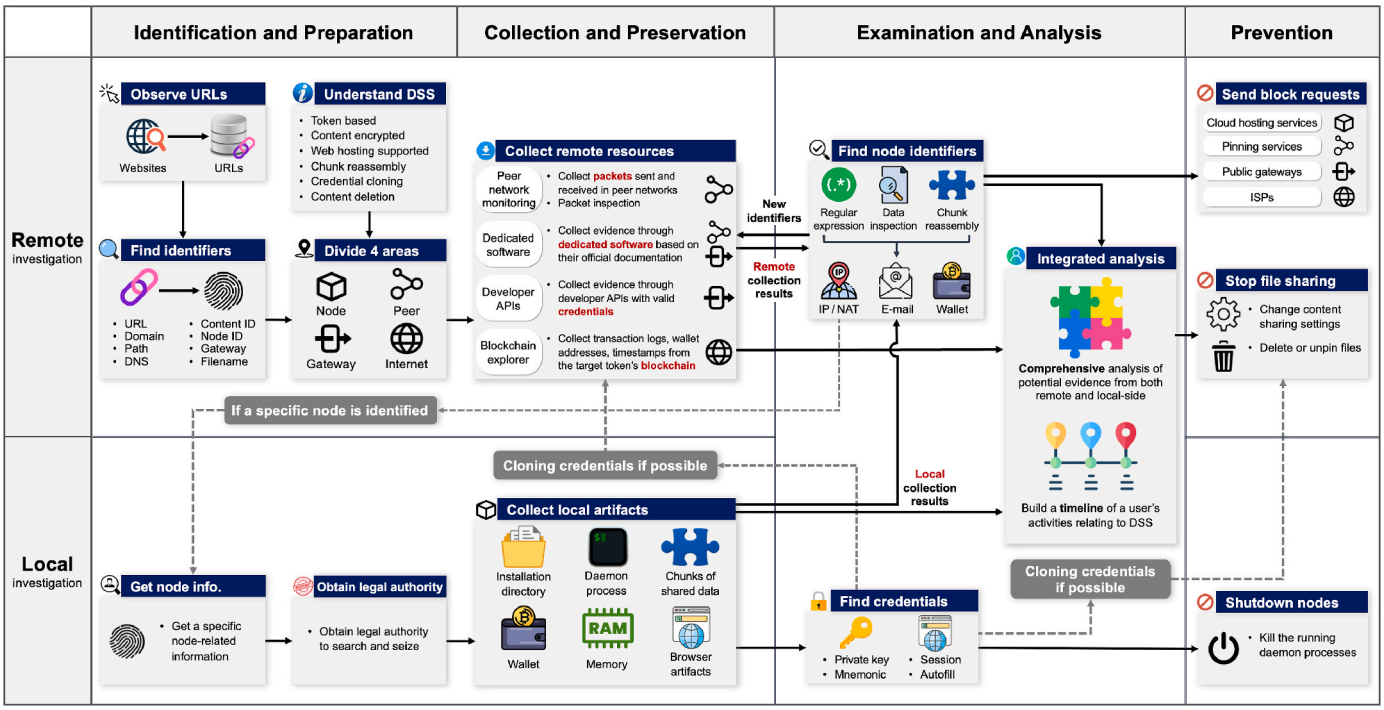

Fig. 2. Flowchart of the proposed forensic investigation framework for decentralized storage services.

2.2. Potential crime scenarios

2.2.1. Phishing

Phishing attacks often use free web-hosting domains due to their advantages such as no investment, ease of creation, secure sockets layer (SSL) certification, premium top level domains (TLDs), and longer domain age (Roy et al., 2022). In addition to these advantages, *IPFS* is becoming a new trend in phishing attacks because content cannot be deleted, registration is not required, and content censorship is not performed. According to Kaspersky, the number of phishing samples per month on *IPFS* has been increasing since November 2022, with more than 300,000 samples found per month (Kaspersky, 2023). Trend Micro has reported that the percentage of phishing samples using IPFS increased from less than 0.5% in the first half of 2022 to about 3% in the second half, and that blocking the *IPFS* gateway is impossible due to NFTs stored in the IPFS network (Trend, 2022).

2.2.2. Copyright infringement

BitTorrent, which has 100 million users, releases *BTFS* and *BitTorrent Speed*, and builds an ecosystem where hosts can receive BTT tokens when they save or share files. Unlike the existing methods of distributing illegal reproductions, illegal profits can be earned by operating a host node in *BTFS* within the BitTorrent ecosystem resulting in large-scale copyright infringement. Z-Library Project, which describes itself as the world's largest e-book library, is using *IPFS* for sharing files. After their website was shut down by the U.S. Department of Justice, they created *Anna's archive*, which provides access through *IPFS*, and is known to have uploaded about 6 million books so far (Anna's blog, 2022).

2.2.3. Malware distribution

The inability to delete content from *IPFS* network creates an ideal environment for malware distribution. According to Patsakis and Casino, the absence of a deletion mechanism in the *IPFS* network makes it effective for distributing malware (Patsakis and Casino, 2019). Recently, Cisco Talos discovered a sample that downloads remote access trojan (RAT) malware from an *IPFS* gateway (Cisco Talos, 2022), and Trend Micro reported malware samples distributed in the *IPFS* network, including infostealers, RATs, and cryptominers (Trend Micro, 2023).

2.2.4. Other crime-related activities

In addition to the representative scenarios mentioned above, there are various cases in which DSS can be associated with forensic investigation. Since DSS is just a type of remote storage that provides privacy and censorship resistance from the client's perspective, it can be an optimal space for criminals to store important or sensitive data while avoiding tracking by investigative agencies.

3. Related works

3.1. Investigation for peer-to-peer file sharing services

Forensic investigation studies on peer-to-peer (P2P) file sharing services can be categorized into local side and remote side approaches. Studies on local side focus on observing the registry, files, and logs during program execution, as well as inspecting the structure of network packets. Acorn Jamie analyzed changes in registry and files for five types of *BitTorrent* client programs (Acorn, 2008). Cannatella and Geoghegan developed a tool that analyze changes in registry and configuration files for *LimeWire* (Cannatella and Geoghegan, 2009). Lallie and Briggs observed how registry changed on Windows 7 for three types of *BitTorrent* client programs (Lallie and Briggs, 2011). Farina et al. analyzed network packets, logs, and registry for the *BitTorrent Sync*'s client program (Farina et al., 2014). Teing et al. proposed an investigative methodology for *BitTorrent Sync*, including the analysis of log files and peer discovery packets (Teing et al., 2017). However, these studies have limitations in that they do not acquire evidence from the remote side.

Studies of P2P file sharing services on the remote side have focused on monitoring network packets exchanged between nodes, and collecting data from a peer network. Schrader et al. developed a tool to monitor *BitTorrent* handshake packets and save them if they match prespecified hash values (Schrader et al., 2009). Bauer et al. monitored the *BitTorrent* swarm through an active monitoring method that can reduce false positives, and developed a framework to identify peers and collect evidence of file sharing (Bauer et al., 2009). Liberatore et al. proposed a

Table 1
Comparison table between seven well-known DSSs in consideration of six forensics-related features.

Features	IPFS	Filecoin based	BTFS	Internet Computer	Storj	Sia	Arweave based
Cryptocurrency	N/A	Filecoin (FIL)	BitTorrent (BTT)	Internet Computer (ICP)	Storj (STORJ)	SiaCoin (SC)	Arweave (AR)
Data encryption	No	No	No	No	Yes	Yes	No
Web hosting	Yes	Depends	Yes	Yes	Yes	No	Depends
Chunk reassembly	Yes	No	Yes	No	No	No	No
Credential cloning	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Content deletion	No	No	No	Yes	Yes	Yes	No

*N/A means there is no cryptocurrency based.

legally valid method for collecting data stored on remote side, based on understanding of how nodes in *Gnutella* and *BitTorrent* communicate within the peer network (Liberatore et al., 2010). Scanlon et al. analyzed the structure of peer discovery and data synchronization packets of *BitTorrent Sync*, and proposed a network investigation methodology (Scanlon et al., 2015). The methodology they proposed involves identifying the secret, finding peer information through peer discovery method, and downloading content. Peersman et al. developed a framework for identifying previously undiscovered child pornography through machine learning based monitoring technique on P2P networks (Peersman et al., 2016). Although these studies are primarily focused on remote side, some studies also take into account local side investigation (Liberatore et al., 2010; Scanlon et al., 2015). However, it is limited to identifying sources of investigation and verifying if content IDs discovered on remote side exist on local system. Moreover, these studies have not adequately considered the gateway area and the internet area.

3.2. Relevant studies on decentralized storage services

Existing studies on DSS has primarily focused on *IPFS*. Patsakis and Casino described the process of creating a botnet using anonymity, persistency, and robust property of *IPFS* network (Patsakis and Casino, 2019). Karapapas et al. also mentioned how ransomware can be distributed using these properties of *IPFS* (Karapapas et al., 2020). Daniel et al. presented a method to identify individual peer nodes in *IPFS* network using metadata of the node and estimated the size of the whole *IPFS* network (Daniel and Tschorsch, 2022). Balduf et al. estimated the size of the *IPFS* network by passively monitoring *BitSwap* messages and suggested three possible privacy attacks on the *IPFS* network (Balduf et al., 2022). They could monitor specific content and nodes, and check whether the specific node has cached certain content or not. The monitoring techniques and privacy attack techniques they suggested were also applied to our case study. These studies contributed to our understanding of the characteristics and monitoring techniques of the *IPFS* network, but they have limitations when it comes to forensic investigations. Specifically, they do not describe in detail how to track a specific node on the remote side, and cannot find evidence of client program execution on the local side. Furthermore, the methodology they proposed was not sufficient to be generalized for the entire DSS. Therefore, a new methodology is needed that takes into account all four areas which comprise the DSS ecosystem, while connecting both local and remote sides.

4. *IF-DSS*: a forensic investigation framework for decentralized storage services

We propose a novel forensic investigation framework for DSS, named *IF-DSS*. As depicted in Fig. 2, the framework consists of detailed steps categorized into the remote and local side of an investigation, depending on the location where potential digital evidence is stored. Building on the traditional digital forensic framework (Kent et al., 2006), our *IF-DSS* framework incorporates the necessary steps to respond effectively to DSSs, including (1) identification and preparation, (2) collection and preservation, (3) examination and analysis, and (4) prevention. Furthermore, we extend the framework by applying the concept that four areas constitute the DSS's ecosystem.

4.1. Identification and preparation

4.1.1. Observing URLs and finding identifiers

We propose that the investigation should begin with an examination of URLs published on the Web. Various identifiers such as content ID, node ID, gateway address, and filename can be extracted from the domain and path of the URL. When the identifier is embedded within the URL, investigators may need to use these services to determine where the specific identifiers exist within the URLs. In some cases, the identifier may be located in the domain name service (DNS) records. For IPFS and Internet Computer, the identifiers can be discovered by querying the *TXT* record of the domain name combined with a service-specific subdomain (DNSLink - IPFS docs; Cutom domains).

4.1.2. Understanding DSS and identifying the four areas

To conduct a comprehensive investigation of user activities on a DSS, its architecture and characteristics must be understood. Each DSS has unique features including incentive payment methods, data encryption techniques, along with data storage and management methods. We identified six key features of DSS that should be considered in forensic investigations and explain how the investigation methodology can differ based on each feature. For a better understanding, we selected seven DSS based on the number of users and cryptocurrency market caps and determined their features, as shown in Table 1. Services based on the Filecoin and Arweave networks are combined into a single column. Although *IPFS* is the underlying protocol of *Filecoin*, it is listed separately owing to its extensive standalone usage. The features outlined in the table are current at the time of writing this paper (May 2023).

- Cryptocurrency

Cryptocurrency serves as a payment method for data storage. Through reading their white papers, investigators can learn the communication methods between clients and hosts and the management techniques for file content and metadata. Additionally, investigators should familiarize themselves with the supported wallets and related blockchain explorers.

- Data encryption

This refers to whether data transmitted from the client to the peer area are encrypted by default. If the data are encrypted, they cannot be decrypted on the peer network or the host node without valid credentials. Consequently, when investigating DSS that offer encryption by default, such as Storj and Sia, investigators must focus on obtaining credentials from the local side. However, if data are not encrypted on the client node, content and metadata can be retrieved from both the peer network and host nodes.

- Web hosting

This indicates whether the content uploaded to a DSS can function as a web page. Because DSS offers censorship resistance and low costs, numerous websites are hosted on web-hosting-supported DSSs. For

instance, because *IPFS* has no hosting costs and supports web hosting, websites such as *Tornado Cash*, which provide money laundering services, are hosted on its network. Web-hosted content is generally linked to other content required for operating web pages, and the DNS records of these websites often retain identifiers.

- Chunk reassembly

Files uploaded on DSS are divided into chunks of a specific size and cached locally in some DSSs, potentially allowing for data restoration by reassembling the divided chunks. Therefore, it is crucial to understand the chunks' structure for each service and prepare reassembly tools. For example, the cached chunks of *IPFS* and *BTFS* can be reassembled into the original files.

- Credentials cloning

This refers to obtaining user credentials on the local system and replicating them on another system. Investigators must know how to acquire user credentials, clone them, and prepare related equipment in advance to search and seizure. Credential cloning was feasible for all the services considered.

Content deletion

Due to the decentralized nature of DSS, content may not be deleted. In *IPFS*, even if the content is deleted from the node and gateway areas, it may remain in the peer area. In *Arweave*, content is stored in *Blockweave* and cannot be permanently deleted (Williams et al., 2019). For these services, investigators must focus on methods to prevent the further distribution of content, as our framework suggests.

4.1.3. Obtaining node information and legal authority

To search and seize a local node, information that uniquely identifies the node must be acquired and its legal authority also be obtained. Based on the node identifiers obtained in Section 4.3.2, investigators can identify the suspect operating a specific node or the physical location at which the node exists.

4.2. Collection and preservation

4.2.1. Collecting remote resources

Using identifiers found in public URLs, potential evidence on the remote side can be collected using four methods. The *peer network monitoring* method captures and inspects communication packets between peer nodes. Investigators can connect to the entire peer network by creating a node that is connected to a large number of peer nodes or by directly connecting to the target node and monitoring the packets received. These packets are generally encrypted, making it difficult to understand the embedded messages. However, investigators can obtain these messages by identifying unencrypted packets, changing the logging configuration of the dedicated software, or by modifying the source code of the dedicated software to log messages.

The dedicated software collects information through commands from official tools provided by the DSS provider. This software can interact with peer networks or gateways, providing various functions such as querying the DHT, establishing connections with other nodes, and downloading content. The tool's documentation offers an in-depth explanation of these functions. For example, *Kubo*, a dedicated software application *IPFS*, can determine the IP address of a content provider node by querying a specific content ID. The dedicated software of *Akord*, an *Arweave*-based DSS, can determine a user's email address via an account ID.

By utilizing the *Developer API* provided by the DSS, information stored in the gateway area, such as user account details, wallet addresses, transaction IDs, content lists, and timestamps, can be obtained. In general, developer APIs can access more information than service websites. Valid user credentials are typically required to use these APIs; however, not all DSS follow this rule. For example, the developer API of *Web3.storage*, a Filecoin-based DSS provides metadata of any shared content, including the contract ID, host node ID, and creation timestamp of the content. The developer API of *Akord*, an Arweave-based DSS, can, without any credentials, retrieve a list of all the files and transaction records associated with an account through the account ID.

Blockchain explorer refers to a website that allows the viewing of transaction logs disclosed on the blockchain. Investigators can access wallet addresses or timestamps by querying the node ID or transaction ID in the cryptocurrency's blockchain explorer. For example, files uploaded to Arweave and their metadata can be downloaded from its explorer website.

4.2.2. Collecting local artifacts

Potential evidence on the local side can be acquired from the artifacts on local devices. Given that data resilience is the primary reason for using a DSS, the initial step involves inspecting the installation directories of all DSSs. This is crucial because multiple DSS nodes can be installed on a single device. The installation directory is created by default and stores important values such as configurations, private keys, and node IDs when dedicated software is installed. Additionally, investigators can check the list of running processes and dump DSS-related process memory. Artifacts related to wallet applications and cached file chunks, if present, must also be acquired. Finally, because there may be evidence on DSS-related websites, browser-related artifacts should also be acquired.

4.3. Examination and analysis

4.3.1. Finding credentials

Credentials refer to the user authority required to access a DSS. Web browser-related credentials can be retrieved from the web browser storage, including Local Storage, Cookies, and Autofill. Node-related credentials include the node's private key, the access token for the metadata server, and the mnemonic code for accessing the cryptocurrency wallet. These credentials are generally stored in the installation directory of the dedicated software, allowing credential cloning by simply copying the entire directory to another system. Credential cloning was feasible for all the seven DSS examined in this study.

4.3.2. Finding node identifiers

In this step, node identifiers are extracted from the data acquired in the previous steps on both remote and local sides. The first method involves searching for known patterns in strings or binary files using regular expressions. Because peer network monitoring logs and process memory dumps can be very large, it is recommended to determine the form of the identifiers and search for identifiers based on regular expressions. The second method involves inspecting the data using the investigator's knowledge and insights. Given the different types of data collected from various sources, investigators must examine these data to determine whether they contain important evidence. If repetitive patterns are observed, the development of regular expressions or automated scripts should be considered. The third method involves reassembling the cached file chunks obtained on the local side. To understand the structure of the file chunks, investigators can search the white papers of the cryptocurrency and the DSS's documentation. In Section 5, we describe how to reassemble the data chunks of *IPFS*. By reassembling the file chunks, the content uploaded or downloaded by the user can be identified. Using the methods mentioned above, node identifiers, such as public and private IP addresses, email addresses, and wallet addresses, can be identified. If a new identifier is found during this process, a recursive search should be performed using collection methods described in Section 4.2.1.

4.3.3. Integrated analysis

Based on the evidence collected from both remote and local sides, investigators could track and reconstruct DSS-related user activities. User events discovered from various sources could be normalized, integrated, and analyzed along with the time values. The investigators could evaluate whether the evidence collected in each of the four areas is aligned with their these hypotheses about user behavior. Through this comprehensive analysis, user activities can be effectively reconstructed.

4.4. Prevention

Preventing the further distribution of illegal content is a critical concern for investigators. Content uploaded to DSSs can be accessed through both peer networks and the surface web, facilitating the sharing and dissemination of content. Therefore, we have added a prevention process to the traditional digital forensic framework (Kent et al., 2006) and present three detailed methods. Because these methods can change the state of potential evidence stored in the DSS, investigators can optionally perform the prevention process after completing all data collection procedures. Thorough consideration is required to determine whether or not this process will be performed and it must be done only after obtaining prior permission from law enforcement executives.

4.4.1. Sending block requests to remote servers

Investigators can send content-blocking requests to each of the four areas of the DSS to block access to content within each area. In the Internet area, the prevention method involves reporting a list of URLs to Internet service providers (ISPs). The criteria for content filtering may vary depending on the country in which they operate (Zittrain and Palfrey, 2008). The result of being filtered by an ISP is that the content hosted by the blocked URLs cannot be accessed from any of the IP addresses in the range provided by the ISP.

In the gateway area, the prevention method involves sending abuse reports to gateway servers. *IPFS*'s official gateway makes efforts to block malicious content by receiving abuse reports from users (IPFS abuse report), but *Storj*'s official gateway makes no effort to block illegal content (Storj's offical blog, 2014). Once blocked by a gateway, such content can no longer be accessed by the specific URL which is hosted by the gateway.

In the peer area, the prevention method involves sending abuse reports to the companies called *pinning service*, which guarantee the storage of the entire file. This method is applicable only to DSS where pinning services exist. Pinning illegal content is prohibited by the pinning service's terms and conditions, and in case of a violation, users may be prohibited from using the service. Because pinning services communicate with many nodes and store a vast amount of content, the probability of finding content is reduced because of being unpinned by pinning services.

Prevention in a node area involves reporting a list of the identified node IP addresses to cloud-based hosting services or ISPs. In their terms and conditions, cloud hosting services such as the Amazon Web Service (AWS) stipulate that their resources cannot be used for illegal purposes. If these terms are violated, the server can be stopped or the content can be deleted. Moreover, if the IP addresses assigned by the ISP are abused to distribute illegal content, the investigative agency can legitimately request that the ISP not assign an IP to the user.

4.4.2. Stopping file sharing

Using the credentials obtained from the local side, investigators can access the suspect's node and the website account. Depending on the features provided by the DSS, investigators can change the settings to stop sharing the content, delete the uploaded content, unpin the content to prevent further distribution from the peer network, or terminate the file storing contract to cease its storage.

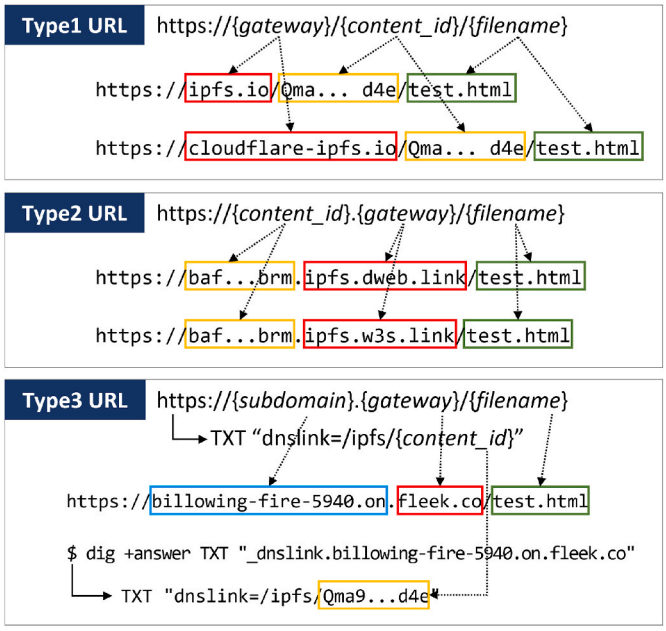

Fig. 3. Finding identifiers from URLs generated by the IPFS gateways.

4.4.3. Shutting down the nodes

When performing live forensics on the local node, if DSS-related daemon processes are found running, investigators can either terminate the detected processes or isolate the node from the network to prevent further sharing of the content.

5. Implementation and case studies for investigating IPFS with filecoin

In this section, we present two case studies to demonstrate the effectiveness and applicability of the proposed *IF-DSS* framework. These case studies focused on *IPFS*, which has the most users among DSS, and *Filecoin*, a service that offers a file storage contract based on *IPFS*.

In the first case study, we discovered identifiers from phishing URLs found on the web and located the IP addresses of the content provider nodes using the four collection methods outlined in the proposed framework. We then sent block requests to prevent further content sharing across all four areas that comprise the DSS's ecosystem.

In the second, we investigated the *IPFS* host node used for large-scale illegal file sharing. In this case study, the target *IPFS* node was used to share files from the *distributed-wikipedia-mirror* (IPFS's Distributed Wikipedia Mirror Project), and file storage contracts were signed with Filecoin-based services, such as *Web3.storage* and *Fleek*. Assuming that the investigators had access to the suspect's PC, where the target IPFS node was running, we acquired local artifacts, reassembled cached chunks, and acquired remote resources using cached credentials. We performed a comprehensive analysis and prevented the further sharing of data on both sides.

5.1. Implementation and dataset

For these case studies, we made the dataset available online, wrote an investigation manual, developed scripts to automate some steps, and made them publicly available (IF-DSS Github). We used a known online dataset available online for the first case study. We also created a dataset by operating an IPFS host node and uploading large-scale files for the second case study. Furthermore, the prepared investigation manual included specific commands, scripts, and detailed procedures used at each stage of both case studies. The developed scripts helped to extract identifiers from *IPFS*-related URLs and query them on the peer network

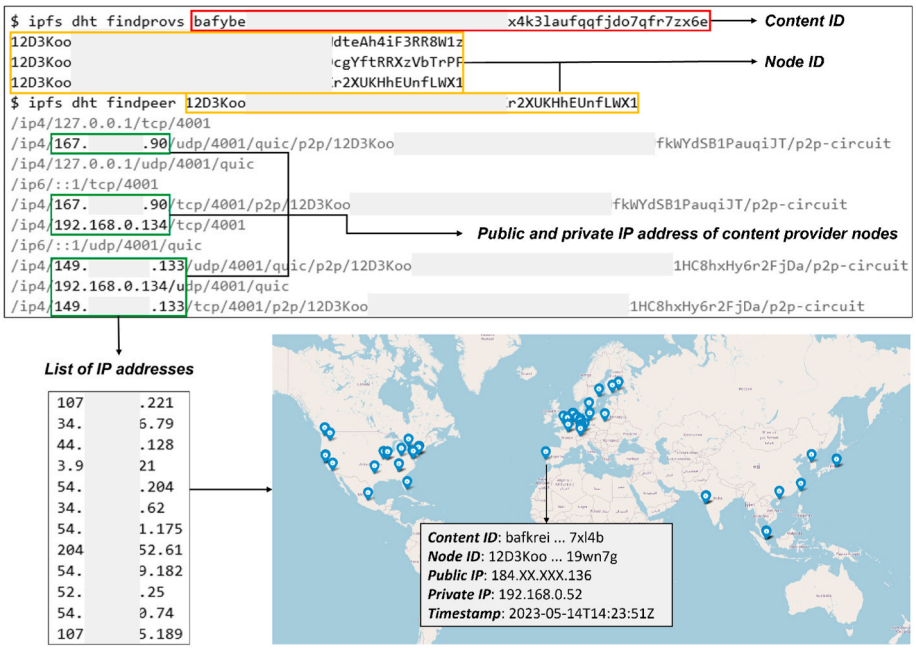

Fig. 4. Identifying IP addresses being used as content provider nodes.

to identify the IP address of the content provider node. Moreover, these scripts automated the processes of deserializing, categorizing, and reassembling the fragmented file chunks.

5.2. Case study 1: phishing URLs hosted on the IPFS network

5.2.1. Identification and preparation

We obtained 61,528 live phishing URLs using *PhishTank* (Phishtank) and identified 3662 as *IPFS*-related URLs (downloaded on April 25, 2023). As shown in Fig. 3, even for the same file, the formats of the phishing URLs varied depending on the gateway server used. These URLs generally contain a Content ID (CID) assigned to the shared file, the gateway name, and the file name. A CID typically begins with `Qm` or `baf`. In some URLs, CIDs could be identified by querying *IPFS*'s DNS records, named *DNSLink* (DNSLink - IPFS docs).

5.2.2. Collection and preservation

By using the discovered CIDs along with *IPFS*'s dedicated software *Kubo*, information about the content provider node could be obtained. We utilized the *findprovs* command, which reveals a node ID being used for distributing specific content, and the *findpeer* command, which provides information about public and private IP addresses associated with a specific node ID. Consequently, those commands allowed forensic examiners to determine the IP-address-based location of the target node used for distributing phishing content, as shown in Fig. 4.

To apply the passive-monitoring technique proposed by Balduf et al. (2022), the *ConnMgr* configuration of an *IPFS* node, which was prepared for the investigation, was changed to remove the limit on the maximum number of nodes that could be connected (Kubo config). Additionally, to collect the *Bitswap* transactions, a message-based protocol for data block exchanges in *IPFS*, the logging level of the node's *engine* subsystem was configured as *debug* (Bitswap - IPFS docs). Consequently, the relevant *Bitswap* messages transmitted by peer nodes could be acquired for examination.

As mentioned in Section 4.2.1, if the target URL's gateway is *Web3.storage*, any shared files' metadata can be retrieved without credentials. The available metadata contain *Filecoin* contract-related identifiers such as miner IDs and deal IDs. Furthermore, *Filecoin* explorers are useful for collecting miner IDs, transaction logs, and timestamps by querying the CIDs and deal IDs (Glif explorer; Filecoin CID Checker; cid.place).

5.2.3. Examination and analysis

We developed regular expressions to filter *BitSwap* messages from the previously acquired monitoring logs. This enabled us to monitor the content that a specific node was attempting to download. Then, using the developed scripts, we removed duplicates from the list of IP addresses used for the content provider nodes and distinguished between public and private IPs. Additional identifiers such as miner IDs and deal IDs were also discovered in the data acquired from the developer APIs and *Filecoin* explorers. These new identifiers were used for further collection.

From the analysis of phishing URLs based on the collected data, we found the following facts regarding the phishing URLs. First, the *IPFS* nodes used to share the phishing content were mostly related to one or more phishing URLs. Second, the provider nodes were spread worldwide, as shown in Fig. 4. Third, 62 of the 3662 phishing URLs were hosted through cloud-based hosting services, such as AWS. Finally, phishing URLs were being distributed through various gateways, not just the official *IPFS* gateway.

5.2.4. Prevention

To prevent further content sharing, content-blocking requests were sent across all four areas of the DSS. For the Internet area, we asked an ISP, to which discovered *IPFS* node belongs, to block several phishing URLs. However, the ISP did not respond in a timely manner. Given the brief life cycle of phishing sites, it was hard to expect the block request to be properly processed. Simultaneously, we requested the *IPFS*'s official gateway to block the malicious content in the gateway area, the content was blocked after a day. For the peer and node areas, we submitted block requests to *Pinata*, one of the biggiest pinning services, and AWS, a cloud hosting service, respectively. As a result, *Pinata* did not respond, but AWS processed the block request in just one day.

5.3. Case study 2: an IPFS host node used for large-scale illegal file sharing

5.3.1. Identification and preparation

In this case study, we prepared an *IPFS* node hosting *distributed-*

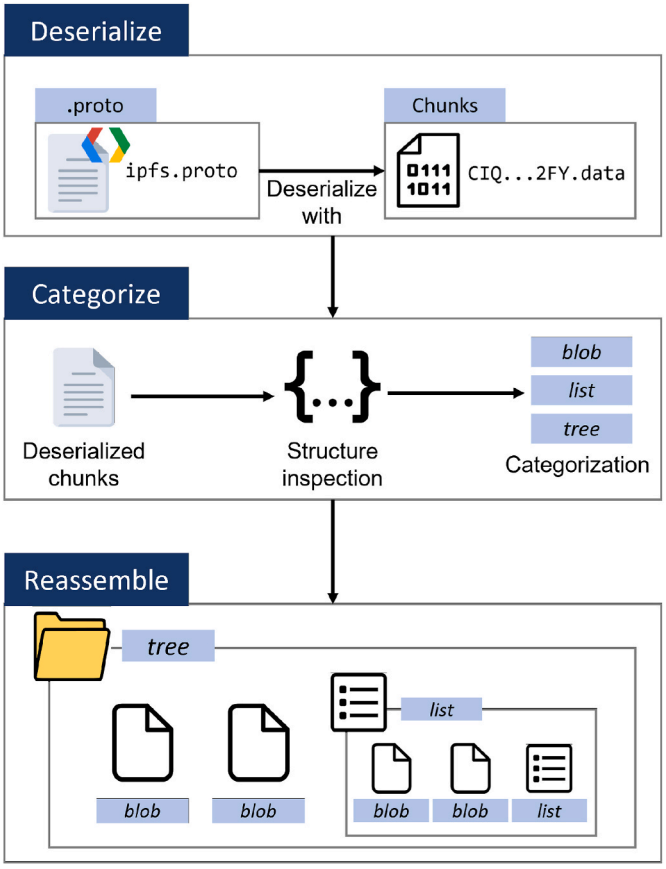

Fig. 5. Reassembling the file chunks cached by the IPFS client node.

wikipedia-mirror. Assuming this to be illegal content, we conducted a hypothetical digital forensic investigation. We also assumed that the investigative agency identified an IP address and the corresponding physical location of the node using the method described in Section 5.2.1 and then obtained a warrant for the search and seizure of a suspect's local system.

5.3.2. Collection and preservation

Upon live forensic analysis of the suspect's local system, it was found that a *Kubo,* a *IPFS* client program, was installed and the associated process was running. After this discovery, the process's memory was immediately dumped, and all files in the entire installation directory of the identified client application were acquired. Additionally, in the History database among the Chrome-related artifacts, there were several records of visits to *Web3.storage* and *Fleek*, both of which were gateway services of *IPFS*. Because the suspect logged into both services using a GitHub account, we also extracted the *user_session* cookie value of the GitHub account. The acquired credentials were then cloned onto a separate laptop prepared for the investigation. We then logged into both *Web3.storage* and *Fleek* and obtained shared content and associated metadata by crawling information displayed on their web pages and by calling web APIs.

5.3.3. Examination and analysis

When examining the acquired potential digital evidence, we found the node's private key and file chunks in the client application's installation directory. To clone credentials, the entire installation directory was copied to a separate laptop, and a cookie value of the suspect's GitHub account was set up on this laptop. Consequently, we could control the suspect's *IPFS* node and access relevant websites using a laptop with the cloned credentials.

To analyze the collected *IPFS* file chunks, we developed a process for reassembling them, as illustrated in Fig. 5. The first step was to deserialize each file chunk, which is stored in the *Protocol Buffers* format (IPLD specs). The next step was to categorize the individual deserialized chunks into one of the following types: `blob`, `list`, or `tree`. These categories are based on the structural details found in the official white paper *IPFS* (Benet, 2014): (1) a `tree-type` chunk is transformed into a directory; (2) `list-type` chunks should be reassembled by merging relevant chunks into a single file, and (3) a `blob-type` chunk is either part of a list or a separate file. By following the above processes, we extracted the original files associated with the CIDs of the *distributed-wikipedia-mirror*. Interestingly, we also identified additional illegal files that had been shared on the node. From the acquired process memory dump, we found *Kubo*'s debug messages, which contained traces indicating that these files had been shared.

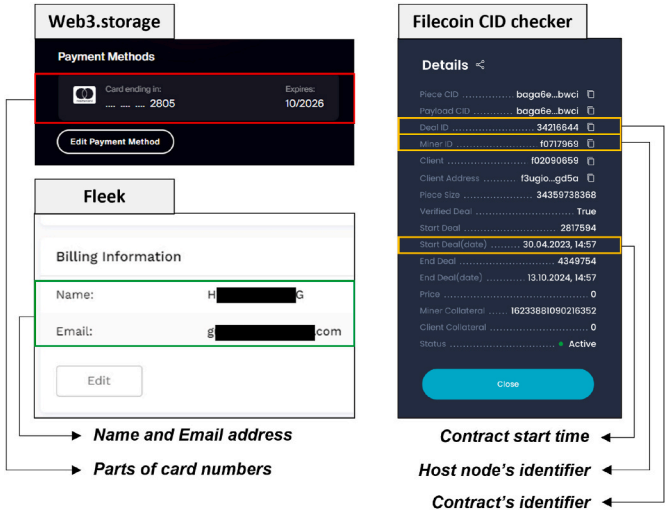

Fig. 6. Information from two IPFS gateways and the Filecoin explorer.

Accessing through the cloned credentials, we obtained an email address of the suspect and a parts of registered card numbers from *Web3.storage* and *Fleek* websites as illustrated in Fig. 6. Additionally, CIDs of illegal files identified through the chunk reassembly were used as search keywords in the *Filecoin* explorer (Filecoin CID Checker). This allows forensic investigators to acquire *Filecoin*-related evidence such as miner IDs and relevant contracts' timestamps, as depicted in Fig. 6.

The potential evidence gathered from various sources was normalized and stored in an integrated database. This enabled effective analysis of user activities related to the DSS, through the creation of timelines and the visualization of information.

5.3.4. Prevention

After selectively collecting data through live forensics on the suspect's local system, running processes related to the DSS were terminated to prevent further sharing of illegal content. Then, all files uploaded to *Web3.storage* and *Fleek* were deleted.

6. Discussion

6.1. Comparative study with the existing frameworks

In this section, we aim to outline potential limitations that might arise when applying existing investigative methodologies, designed for cloud-based storages and P2P-based file sharing services, to the investigation of DSS. Since there are no previous studies that directly address investigative methodologies for DSS, we reviewed literature dealing with investigation methodology for services that store data in remote locations. We selected literatures based on their target service's similarity to DSS and their citations, while excluding one that only presented

Table 2
Summary of comparative study with the existing digital forensic investigation frameworks.

Subject	Ref	Year	Preparation	Collection				Examination		Prevention
			Understand four area of the DSS	Node area	Peer area	Gateway area	Internet area	Chunk reassembly	Using credentials	Practical methods for prevention
Cloud	Chung et al.	2012		✓		✓			✓	
	Yang et al.	2022		✓		✓			✓	
P2P	Liberatore et al.	2010		✓	✓					
	Scanlon et al., 2015	2015		✓	✓					
	Teing et al.	2017		✓	✓					
DSS	Balduf et al.	2022			✓	✓				
	IF-DSS	Proposed	✓	✓	✓	✓	✓	✓	✓	✓

a theoretical framework. As a result, two studies related to cloud-based storage services (Chung et al., 2012; Yang et al., 2022), three studies related to P2P-based file sharing services (Liberatore et al., 2010; Scanlon et al., 2015; Teing et al., 2017), and one study on IPFS network monitoring (Balduf et al., 2022) were chosen for comparison. The results of this comparison are summarized in Table 2.

6.1.1. *Four areas of the DSS and collection of potential digital evidence*

None of the studies under review considered all four areas of DSS, that is node, peer, gateway, and Internet area. Studies on cloud-based storage only took into account the node and gateway areas, while studies on P2P-based file sharing services only considered the node and peer areas. A study directly targeting *IPFS* did consider the node, peer, and gateway areas but did not address *Filecoin*, which is associated with the Internet area (Balduf et al., 2022).

6.1.2. *Chunk reassembly*

None of the studies has proposed a method for reassembling the cached file chunks.

6.1.3. *Using credentials*

The studies on cloud-based storage leverage users' credentials obtained from local side to collect data on remote side. However, the other studies do not involve the collection or utilization of credentials.

6.1.4. *Practical methods for prevention*

None of the studies proposed the practical method to prevent further sharing and dissemination of content. Unlike traditional P2P-based sharing services, DSS allows the content to be shared on the surface web via public gateways. Therefore, prevention methods of such content sharing has great importance and should be considered.

6.2. *Limitations of the proposed framework*

The current version of the framework proposed in this study does have several limitations. First, during the remote investigation step, a node information related to URLs containing illegal content may not be necessarily identified. This could be due to the insufficiently disclosed information for identification or due to the insufficient authority to access the information. Second, the prevention methods may not always be effective. For instance, even if some gateways or pinning services successfully block the specific illegal or malicious content, the content may be shared continuously via other services that were not identified during the investigation process. Finally, the proposed framework was designed primarily around *IPFS* and *Filecoin*, and may not reflect all of the operating mechanisms and characteristics of various DSSs.

7. Conclusion and future directions

This paper introduces the four areas constitute the ecosystem of the DSS and suggests the potential crime-related scenarios related to the DSS. Building on these, we proposed *IF-DSS*, a forensic investigation framework designed specifically for the DSS. In our proposed framework, potential evidence is identified, collected, and analyzed on both the local and remote sides. The usefulness and applicability of the proposed framework were demonstrated through two case studies utilizing *IPFS* with *Filecoin*. We have underscored the unique features of *IF-DSS* by comparing it with the existing investigative methodologies for the cloud-based storage sevices and the P2P-based file-sharing services. We also discussed the limitations of the current version of the framework.

Given the increasing use of the DSS for the sharing of illegal content recently, it is crucial to develop and refine forensic investigation techniques to effectively counteract malicious activities in these environments. Future research should focus on addressing the limitations of the current framework, automating repetitive processes, and verifying the framework's applicability to emerging DSSs. By continuously improving our understanding of the DSS and adapting our investigative methods accordingly, we hope to contribute maintaining the ability to investigate potential digital evidence, regardless of where it is located.

Acknowledgements

This work was supported by a Korea University Grant, and also supported by Police-Lab 2.0 Program(www.kipot.or.kr) funded by the Ministry of Science and ICT(MSIT, Korea) & Korean National Police Agency (KNPA, Korea) [Project Name: Research on Data Acquisition and Analysis for Counter Anti-Forensics/Project Number: 210121M07].

References

Acorn, J., 2008. Forensics of Bittorrent. Tech. rep. Technical Report RHUL-MA-2008-04.
Anna's archive. https://annas-archive.org/. (Accessed 14 May 2023).
Balduf, L., Henningsen, S., Florian, M., Rust, S., Scheuermann, B., 2022. Monitoring data requests in decentralized data storage systems: A case study of IPFS. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 658–668.
Bauer, K., McCoy, D., Grunwald, D., Sicker, D., 2009. Bitstalker: Accurately and efficiently monitoring bittorrent traffic. In: 2009 First IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, pp. 181–185.
BitTorrent Tracker. https://github.com/webtorrent/bittorrent-tracker. (Accessed 14 May 2023).
Cannatella, J., Geoghegan, S.J., 2009. PHAT: a P2P history analysis tool. Journal of Computing Sciences in Colleges 24 (5), 57–64.
Chung, H., Park, J., Lee, S., Kang, C., 2012. Digital forensic investigation of cloud storage services. Digit. Invest. 9 (2), 81–95.
Custom domains. Internet computer. https://internetcomputer.org/docs/current/developer-docs/production/custom-domain/. (Accessed 14 May 2023).
Daniel, E., Tschorsch, F., 2022. Passively Measuring IPFS Churn and Network Size. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, pp. 60–65.
Filecoin price. https://coinmarketcap.com/currencies/filecoin/. (Accessed 14 May 2023).
Filecoin CID Checker. https://filecoin.tools/. (Accessed 14 May 2023).
Bitswap - IPFS docs. https://docs.ipfs.tech/concepts/bitswap/. (Accessed 14 May 2023).
IPFS abuse report. IPFS. https://ipfs.tech/help/. (Accessed 14 May 2023).
DNSLink - IPFS docs. IPFS. https://docs.ipfs.tech/concepts/dnslink/. (Accessed 14 May 2023).
cid.place. https://cid.place/. (Accessed 14 May 2023).

Farina, J., Scanlon, M., Kechadi, M.-T., 2014. Bittorrent sync: First impressions and digital forensic implications. Digit. Invest. 11, S77–S86. https://doi.org/10.1016/j.diin.2014.03.010 proceedings of the First Annual DFRWS Europe.
Glif explorer. https://explorer.glif.io/. (Accessed 14 May 2023).
IF-DSS. Forensic investigation framework of decentralized storage services. https://github.com/hunjison/IF-DSS.
IPFS's Distributed Wikipedia Mirror Project. https://github.com/ipfs/distributed-wikipedia-mirror. (Accessed 14 May 2023).
IPLD specs. dag-pb spec. https://ipld.io/specs/codecs/dag-pb/spec/. (Accessed 14 May 2023).
Karapapas, C., Pittaras, I., Fotiou, N., Polyzos, G.C., 2020. Ransomware as a service using smart contracts and IPFS. In: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE, pp. 1–5.
Kent, K., Chevalier, S., Grance, T., 2006. Guide to integrating forensic techniques into incident. Tech. Rep. 800–886.
Lallie, H.S., Briggs, P.J., 2011. Windows 7 registry forensic evidence created by three popular BitTorrent clients. Digit. Invest. 7 (3), 127–134. https://doi.org/10.1016/j.diin.2010.10.002.
Liberatore, M., Erdely, R., Kerle, T., Levine, B.N., Shields, C., 2010. Forensic investigation of peer-to-peer file sharing networks. Digit. Invest. 7, S95–S103.
Patsakis, C., Casino, F., 2019. Hydras and IPFS: a decentralised playground for malware. Int. J. Inf. Secur. 18, 787–799.
Peersman, C., Schulze, C., Rashid, A., Brennan, M., Fischer, C., 2016. iCOP: Live forensics to reveal previously unknown criminal media on P2P networks. Digit. Invest. 18, 50–64.
Phishtank. https://phishtank.org/developer_info.php. (Accessed 14 May 2023).
IPFS docs. IPFS.https://docs.ipfs.tech/how-to/command-line-quick-start/. (Accessed 14 May 2023).
Set up - Filecoin docs. https://lotus.filecoin.io/tutorials/lotus/store-and-retrieve/set-up/. (Accessed 14 May 2023).
Public gateway checker of IPFS. https://ipfs.github.io/public-gateway-checker/. (Accessed 14 May 2023).
Storj's Satellite. https://docs.storj.io/dcs/concepts/satellite. (Accessed 14 May 2023).
Scanlon, M., Farina, J., Kechadi, M.-T., 2015. Network investigation methodology for BitTorrent Sync: A Peer-to-Peer based file synchronisation service. Comput. Secur. 54, 27–43.
Schrader, K., Mullins, B., Peterson, G., Mills, R., 2009. Tracking contraband files transmitted using BitTorrent. In: Advances in Digital Forensics V: Fifth IFIP WG 11.9 International Conference on Digital Forensics, 5. Springer, Orlando, Florida, USA, pp. 159–173. January 26-28, 2009, Revised Selected Papers.
Storj's Canary. https://www.storj.io/canary.txt. (Accessed 14 May 2023).
Teing, Y.-Y., Dehghantanha, A., Choo, K.-K.R., Yang, L.T., 2017. Forensic investigation of P2P cloud storage services and backbone for IoT networks: BitTorrent Sync as a case study. Comput. Electr. Eng. 58, 350–363.
Kubo config. https://github.com/ipfs/kubo/blob/master/docs/config.md#swarmconnmgr. (Accessed 14 May 2023).
Benet J., 2014. IPFS - Content Addressed, Versioned, P2P File System, arXiv preprint arXiv:1407.3561.
Trust Wave, 2022. The new hotbed of phishing. https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/ipfs-the-new-hotbed-of-phishing/. (Accessed 14 May 2023).
Trend Micro, 2023. A new data frontier or a new cybercriminal hideout? https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ipfs-a-new-data-frontier-or-a-new-cybercriminal-hideout. (Accessed 14 May 2023).
Storj's offical blog, 2014. https://www.storj.io/blog/not-on-my-drive-saying-no-to-illegal-content-on-storj. (Accessed 14 May 2023).
Anna's blog, 2022. https://annas-blog.org/putting-5,998,794-books-on-ipfs.html. (Accessed 14 May 2023).
Kaspersky, 2023. Using decentralized file system in their campaigns. https://www.kaspersky.com/about/press-releases/2023_scammers-go-interplanetary-using-decentralized-file-system-in-their-campaigns. (Accessed 14 May 2023).
Trend Micro. 2022. Web3 IPFS Currently Used For Phishing. https://www.trendmicro.com/en_us/research/22/l/web3-ipfs-only-used-for-phishing—so-far.html. (Accessed 14 May 2023).
Cisco Talos, 2022. Cyber criminal adoption of ipfs for phishing, malware campaigns. https://blog.talosintelligence.com/ipfs-abuse/. (Accessed 14 May 2023).
DHT - IPFS docs. IPFS.https://docs.ipfs.tech/concepts/dht/. (Accessed 14 May 2023).
Roy S. S., Karanjit U., Nilizadeh S., 2022. A Large-Scale Analysis of Phishing Websites Hosted on Free Web Hosting Domains. arXiv preprint arXiv:2212.02563.
Williams, S., Diordiiev, V., Berman, L., Uemlianin, I., 2019. Arweave: A protocol for economically sustainable information permanence. arweave. org, Tech. Rep.
Yang, J., Kim, J., Bang, J., Lee, S., Park, J., 2022. CATCH: Cloud Data Acquisition through Comprehensive and Hybrid Approaches. Forensic Sci. Int.: Digit. Invest. 43, 301442.
Zittrain, J., Palfrey J., 2008. Internet filtering: The politics and mechanisms of control. Access denied: The practice and policy of global internet filtering. 41.