Blockchain-Based Distributed Cloud Storage Digital Forensics: Where's the Beef?

Joseph Ricci | University of New Haven and Lodestone Security
Ibrahim Baggili | University of New Haven
Frank Breitinger | University of New Haven

The current state of the art in digital forensics has primarily focused on the acquisition of data from cloud storage. Here, we present a new challenge in digital forensics: blockchain-based distributed cloud storage, using STORJ as a technology example.

Advances in storage technology have made it progressively more difficult for forensic examiners to acquire data during the digital forensic process. No longer are data stored with a single cloud service provider: they are now stored at several locations across large geographic areas on different user machines.

In this article, we discuss research that has been conducted in cloud storage forensics and how it contrasts with distributed blockchain-based cloud forensics. We explain how this technology has been implemented and the potential mechanisms and challenges the technology presents to forensic investigators. In addition, we examine future research directions for overcoming these challenges.

Traditionally, digital forensics has focused on acquiring data from physically available storage media, such as hard disk drives (HDDs), CDs, or secure digital cards. While these media initially presented challenges for forensic examiners, practitioners today understand the technology and its documented file systems, resulting in a mainstream digital forensics process and several mature tools that have helped automate the acquisition, authentication, and analysis of digital evidence in a fairly feasible manner. However, swift technological progress presents new challenges to the cyberforensics community. For instance, solid-state drives (SSDs) are, in many instances, capable of permanently removing data, making it more difficult to recover potentially incriminating evidence. While the acquisition process remains similar to traditional methods, new issues are on the horizon.

Technology companies quickly recognized the extensibility the cloud provides as a storage medium, allowing people and organizations to access their data from anywhere with an always-connected device—moving away from conventional physical storage media. Today, cloud storage services, such as Google Drive, Dropbox, OneDrive, and so forth, are widely adopted and still on the rise. Although they are centralized by the service provider, it is difficult for investigators to recover data because of limitations due to various legal challenges.[1–7] For investigators, the process is no longer as simple as accessing a potential criminal's physical HDD or SSD. Warrants need to be written out to the

Digital Object Identifier 10.1109/MSEC.2018.2875877
Date of publication: 20 March 2019

cloud storage service hosting the information, which contains thousands or even millions of bits of confidential user data.[8] To address this challenge, practitioners and researchers have employed application programming interfaces (APIs) implemented by cloud storage service providers, allowing for the recovery of files and metadata from these services via the Internet, assuming that the investigative party has authenticated legal access to the cloud storage service.

Enter Distributed Blockchain Storage

While centralized cloud storage is currently commonplace, we anticipate a new realm of distributed data storage that employs blockchain technology, in which users rent out their unused disk space to store chunks of other users' data. This technology is not only encrypted by default; the distributed nature of the storage and decentralized nature of the architecture of this service (rather than Google, for example, storing data for one user, now Bob, Alice, John, and Sarah are all storing the user's data, all over the world) pose yet another set of challenges that need to be overcome by the digital forensics community.

In this article, we provide background information about the types of technologies involved in blockchain that have made this distributed storage technology possible. We also share challenges facing forensic examiners and the research that will need to be conducted to overcome them. We argue that forensically sound methodologies and tools are needed for the acquisition of data from blockchain-based distributed storage systems, such as STORJ (pronounced *storage*).

Summary of Data Acquisition Complexity

In Figure 1, we provide a summary of the growing complexity of forensically acquiring data as related to advances in storage technologies. For clarification, a padlock indicates that the storage is encrypted. The easiest of all of the technologies from which to forensically acquire data is arguably local storage devices, such as the HDD, SSD, and memory. The HDD, when unencrypted, contains plaintext data and is readily available for acquisition and analysis. The HDD analysis process becomes difficult when the drive is encrypted and the key or keys for decryption cannot be located. Finding the one or more keys requires further analysis of evidence acquired from the crime scene or from the suspect.

Memory is slightly more difficult to analyze, given that memory does not generally store data in fixed locations, as do the HDD and SSD. Another challenge memory presents is that its data, unlike the HDD and SSD, are not persistent, meaning that they are not stored permanently—and when a computer is turned off, data in memory are erased at different rates due to leakage currents in which bit values are lost over time. This can render data acquisition efforts in memory fruitless. When encrypted data are added to the mix, the process becomes even more difficult.

Cloud services have expanded in the past decade, offering many advantages to consumers and developers. However, for digital forensic practitioners, this has presented more challenges now that data are no longer stored locally but in server farms located in other states or countries. This, alone, makes it difficult for law enforcement to recover the information, given that they now need a warrant asking service providers for access to drives that could potentially contain other user data. It also adds greater complexity to forensic acquisition and requires tools, such as kumodd (discussed in the following section), to pull data through service APIs. Again, when adding encrypted data to cloud storage, the process becomes more difficult, and necessary keys and credentials are required to recover data.

The easiest of all of the technologies from which to forensically acquire data is arguably local storage devices, such as the HDD, SSD, and memory.

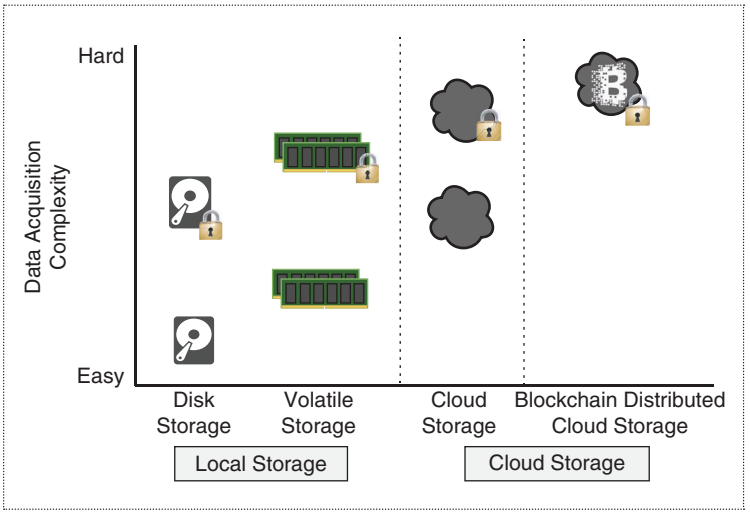

Figure 1. The progressive difficulty and complexity of data acquisition in changing storage technologies.

Last, we come to blockchain storage. By default, all data that are uploaded are encrypted, so there is no reasonable chance of recovering the data without the encryption keys (depending on the strength of an encryption algorithm). Next, data are no longer stored in a central location but dispersed among many user computers that span many geographic areas, making writing a warrant challenging. Not only is it difficult to identify where the data are stored; they do not reside on a single service provider's servers. Even if warrants were written out for each device, the information is encrypted and mixed with other user data. This makes acquisition significantly more difficult because data must be pulled from several storage devices, the correct order of the data must be determined, and keys to decrypt the data must be acquired.

Previous Work

Central cloud storage services have been around for a while, and, as research regarding distributed blockchain cloud services like STORJ do not yet exist, it is necessary to discuss related work. Such services as Google Drive, Dropbox, and ownCloud have been forensically analyzed to provide a methodology and some insight as to how they work. The rest of this section examines related research and services that have been forensically examined, along with the methods used to overcome some of the challenges investigators may encounter.

Bitcoin Forensics: A Tutorial

Past work[9] includes tutorials for Bitcoin forensics, providing knowledge on how to forensically analyze Bitcoin and insights about what is in a blockchain and the transactions made between peers. The work indicated that data contained within blockchain is pseudonymous, meaning anyone can view transactions that occur between individuals, back to the initial transaction. Given that blockchain is an integral component of Bitcoin, the methodology proposed in this tutorial provides additional insight into some of the mechanisms on which blockchain-based storage operates.

A Proposal for a Secure Peer-to-Peer–Type Storage Scheme Using Secret Sharing and Blockchain

A peer-to-peer (P2P) storage network proposed in one study[10] was designed to make it difficult for attackers to target user data in online storage because user data are divided into parts by secret sharing and distributed to P2P nodes via anonymous communication. Even if the state of the P2P network varies over time between storing and restoring operations, the proposed scheme in past work ensures that the user can identify target nodes storing the metadata by utilizing blockchain technology with only memorable secure information for user authentication. While this work proposed a method to secure network storage data through P2P, it also employed blockchain technology to secure metadata and monitor suspicious activity on the network.

Examining Forensic Artifacts Produced by Use of Bitcoin Currency

In one author's dissertation,[11] researchers identify potential evidence by analyzing Bitcoin transactions based on the steps of an effective digital investigation defined by the Digital Forensics Research Workshop: identification, preparation, approach strategy, preservation, collection, examination, and analysis. The research points out that a case is successfully prosecuted based on two key factual areas: the evidence recovered at the scene and the telltale data extracted from the analysis of each piece of digital media. The work explains in detail how Bitcoin transactions work and the artifacts generated on a local machine. Artifacts like Bitcoin addresses, public key and public key hash, transactions, amount, Internet Protocol (IP) addresses, time stamps, and more were found in the local machine's random-access memory, HDD, and Internet browser. As one can see, the bulk of the research conducted on blockchain forensics has not focused on blockchain distributed storage; therefore, we discuss state-of-the-art work in cloud and distributed storage forensics in the sections that follow.

Google Drive: Forensic Analysis of Cloud Storage Data Remnants

Research on Google Drive has identified several artifacts on a virtual machine, such as the username and password.[6] In later work, the authors employed CCleaner and Eraser to simulate antiforensics (deleting data from the devices) and attempted to recover data. Network traffic and memory analysis was also performed to identify critical pieces of data that might assist in reconstructing evidence. Such evidence is critical to forensic investigators, given that it provides them the ability to access the server side of the service.

Cloud Storage Forensics: OwnCloud as a Case Study

Researchers in one study identified artifacts created by the public cloud storage-as-a-service (StaaS) software package ownCloud.[12] This free cloud service is popular among academic institutions and can be implemented on several desktop and mobile platforms. The contributions made by the authors were technical recommendations for the forensic analysis of ownCloud StaaS instances.

The researchers discovered artifacts on the client side of ownCloud, such as sync and file management metadata, cached files, cloud service and authentication data, encryption metadata, browser artifacts, mobile client artifacts, and network analysis. On the server side, they also found such artifacts as administrative and file management metadata, stored files, encryption metadata, cloud logging, and authentication data. While these studies are important for furthering the digital forensics body of knowledge, developing tools to aid in the acquisition of digital evidence from the cloud is of utmost importance to practitioners.

Cloud Forensics: Tool Development Studies and Future Outlook

The authors of another work[13] address the challenge identified in prior research on cloud storage forensics (conducted mostly on the client side). Notwithstanding this focus, web-based StaaS applications are a particularly difficult test for existing forensic tools, which focus almost exclusively on client-centric investigations and examine local storage as the primary source of evidence. Given the nature of StaaS, there is a concern because forensic investigators could potentially miss critical evidence stored on a cloud service. These researchers offered an API-based tool, kumodd, created for data acquisition from Google Drive, Microsoft OneDrive, Dropbox, and Box. The tool can provide an investigator with a list of files stored on a cloud drive. Once the investigator can identify potential evidence files, these can be downloaded in a forensically sound manner for further analysis.

Distributed File System Forensics: XtreemFS as a Case Study

Researchers have also conducted an in-depth analysis of the XtreemFS as a case study for distributed file system forensics.[14] XtreemFS splits file data and replicates them across several storage servers. The authors utilized their cloud forensic framework to conduct their investigation. They were able to recover artifacts unique to the distributed file system: volatile environment data, such as logical network location; nonvolatile environment metadata, such as logging and backup data; and configuration files, such as authentication, network, and operational information.

Forensic Investigation of P2P File-Sharing Networks

Other investigators[15] detailed the functionality of the P2P networks Gnutella and BitTorrent and described the legal challenges involved in investigating such protocols. The authors developed the tool RoundUp for Gnutella probes, following research conducted for network investigations. RoundUp was created to assist sleuths in recovering artifacts, such as files of interest, a peers self-report IP address, and publicly available IP addresses, if a firewall is implemented and displays the push proxies.

While it is important to investigate cloud-based storage, a new storage technology based on blockchain has been emerging, an implementation example of which is STORJ. In the rest of this article, we focus on STORJ to provide the community with a deeper understanding of how the technology works along with the forensic challenges it presents.

Background on STORJ

STORJ[16] is a P2P cloud storage network implementing end-to-end encryption and allowing users to transfer and share data without reliance on a third-party provider, such as Google Drive. One may think of the STORJ system as the "Airbnb of storage," where users rent unused drive space to other users on the network. It is an open-source project designed to create distributed cloud storage by employing a Satoshi-style blockchain, commonly used in cryptocurrencies. However, because blockchain and STORJ are relatively new, little research has been conducted to identify the forensic artifacts left on a user's system. It is imperative for digital forensic examiners to identify client-side artifacts and methods of recovering them, as in Bitcoin investigations.[17] Blockchain distributed storage is different from traditional cloud storage services because it utilizes the disk space of other users on the STORJ network. Instead of an individual's file being uploaded to a central cloud storage provider, the file is divided into smaller pieces called *shards*, which are sent to multiple computers on that network.

While it is important to investigate cloud-based storage, a new storage technology based on blockchain has been emerging, an implementation example of which is STORJ.

To help readers understand STORJ operation, we first share a basic explanation of how the network locates files using Kademlia distributed hash tables (DHTs), what blockchain is, how it works, and how it applies to distributed cloud storage. This background

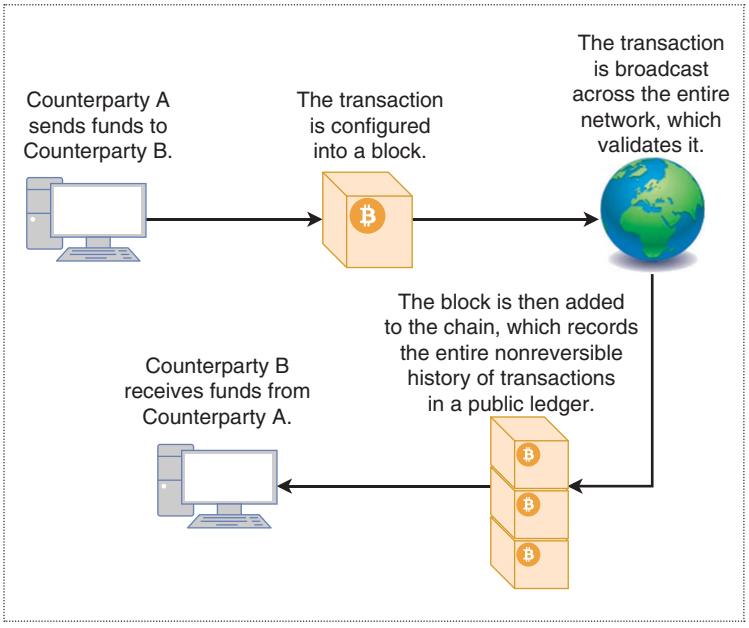

Figure 2. An overview of how blockchain works.

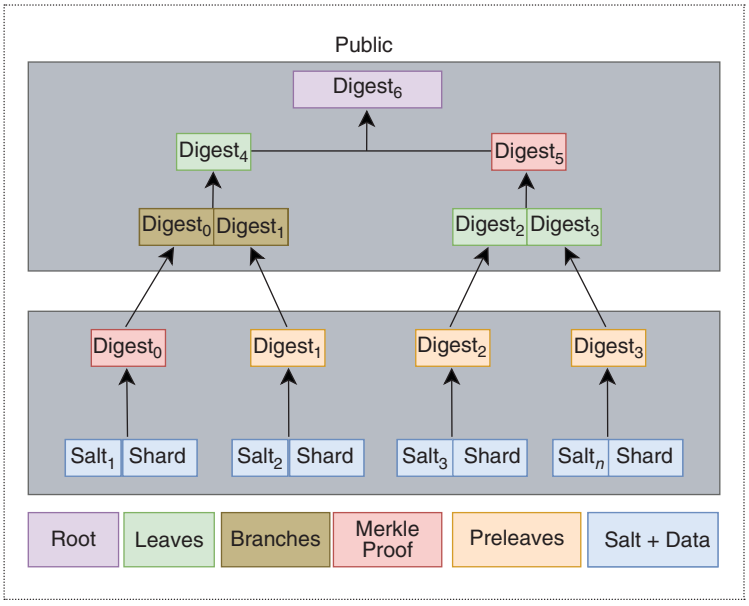

Figure 3. A simple example of a Merkle tree.

regarding these two technologies will provide the reader with an idea of the challenges facing the digital forensics community.

Kademlia

Kademlia is a DHT used for decentralized P2P computer networks. It allows users to quickly upload or download files to or from several computers (also known as *nodes*) on the network through a series of lists unique to each node. Each node has its own ID and a list of the other nodes on the network. The node ID serves not only for identification; the Kademlia algorithm also uses the node ID to locate values (usually file hashes or keywords). In fact, the node ID provides a direct map to file hashes and stores information about where the file or resource can be obtained.

The node lists contain several node IDs, IP addresses, and ports that point to other nodes on the network so that a hash table is distributed across several nodes on the network. In this way, no single server contains all the nodes with the location of files and nodes on the network, dividing responsibility for maintaining the DHT.

Blockchain

Blockchain, a relatively new technology, is a public ledger used to keep track of transactions for digital currencies like Bitcoin[18] and Litecoin. Its power lies not only in its heavy encryption but also in its distribution across a chain of computers, rendering it even more difficult (and prohibitively expensive) to attack. Each time a transaction is made employing one of these currencies, a set of data confirming that a transaction has taken place is saved in the blockchain. Multiple transactions make up a block, which is saved chronologically to a series of other blocks, creating a blockchain (Figure 2).

Data in a transaction include the file hash, network locations of the shard copies, and Merkle roots (explained in the following paragraph). This is where the network can achieve consensus on file location and integrity because every time a negotiation occurs, a message is broadcast throughout the network and verified by nodes on the network.

There are several mechanisms in place to verify that the integrity of a shard has not been compromised. The first is to use Merkle trees,[12] Merkle proofs, and pregenerated audits. A Merkle tree is made up of several paired hashes of data. When these hashes are combined, they form a leaf. These leaves are continually combined until they all become a single hash known as the *Merkle root* (Figure 3). A Merkle branch is made up of one or more leaves that connect directly to the Merkle root. The authors[12] claim that, given the premise of cryptography and hashing, the heartbeats (a standard format for issuing and verifying proofs of retrievability via a challenge–response interaction) cannot be brute-forced. The hash responses can be verified via the Merkle root, which is inserted into a blockchain.

STORJ

Given that most Internet-connected computers have unused hard-drive space, users may sell this excess disk space on the network. Files on this platform will

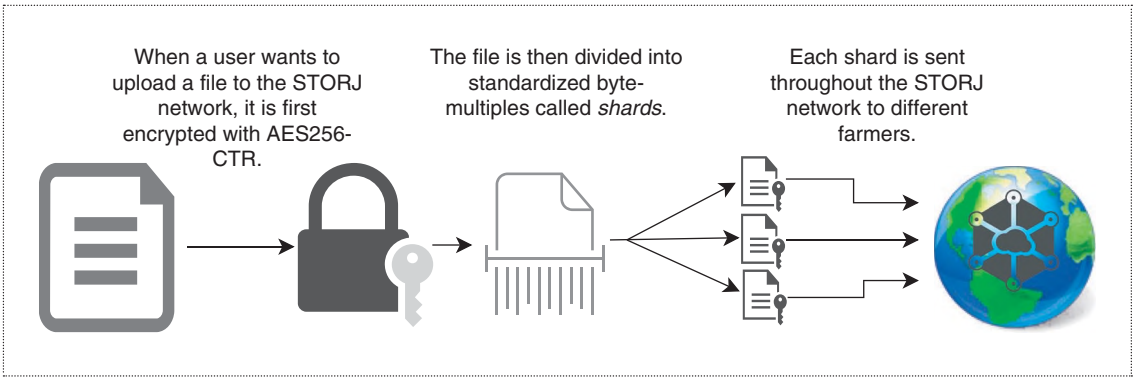

Figure 4. An overview of the client-side process.

be handled by their content via a hash. If one copy of the file is still available on STORJ, the data on the network would be resistant to censorship, tampering, and/or data failure.

The process is as follows. First, a file is encrypted using SHA256-CTR. Then it is split into smaller, standardized byte-multiples (either 8 or 32 MB) to anonymize the file. File sizes or multiple smaller files are combined to form a shard (a small portion of the original file that is encrypted), and any extra space is padded. Simply put, each shard is then salted, hashed, and transmitted to the network (Figure 4).

To explain this in greater detail, prior to being transmitted to the STORJ network, an agreement must first be made between a STORJ client and as many STORJ farmers as needed (depending on the file size and redundancy required). This agreement will also depend on the quality of the storage service provided by the farmers. STORJ can identify which farmers are employing the fastest servers and set a price. The client can bid for that service if it is what is wanted and can pay a premium price for a faster server. The reasoning is that, if a client wants to upload and stream a video, it would be ideal to work with a faster server. In contrast, if a client just wants to perform backups, a typical computer or laptop storage would work and be a cheaper and more efficient solution. After deciding what works best for the client, an agreement is made between the client and the farmer.

Challenges for Examiners

As we have shown, technology has progressed and presents its own challenges to those in the digital forensics arena. Recovering data from HDDs, SSDs, or the cloud has been difficult but not impossible. Automated tools help the process, and investigators recover data with relative ease. Methods have been researched and put to use to recover data with each advance in storage technology. Acquisition of the physical drives is still necessary, but analysis of the devices provides a high level of certainty that the evidence found on them is authentic. Yet, as we begin to look at STORJ, we realize that it is an entirely different beast. From a high-level perspective, we are still dealing with computers and disk drives; but, digging deeper into the technical details, we must consider the complexity and scalability of STORJ.

Digital forensics in this case no longer involves simply acquiring images from disk drives but rather requires utilizing an API to recover data. With STORJ, multiple facets need to be considered to conduct a complete forensic examination. If a file is not recoverable on the suspect's local machine, then the data are in the hands of the intricate mechanisms of STORJ's network. We see some of the challenges as follows:

- recovering files from the local storage
- recovering deleted files
- decrypting encrypted files
- interfacing with STORJ's API
- acquiring user credentials
- acquiring a passphrase
- locating shards
- recovering shards
- warrants to recover shards
- legal jurisdictions related to data storage
- shards that are encrypted and salted
- shards that are combined with other shards
- identifying a shard's geographical location.

When acquiring data from local storage, it would be ideal if potentially incriminating data were left untouched and unencrypted, making the acquisition process simpler. Even if the data were encrypted or deleted, the process could arguably be easier to recover digital evidence compared to other storage technologies. However, when this is not the case and data are not recoverable from the local storage, it is then necessary

to determine what services could have potentially been used to store the data. For example, cloud services have become commonplace for data storage. A user logs into an account through the service provider's platform or web interface and can observe files with a simple user interface. With STORJ, however, data are not simply uploaded to a server, as noted previously. They are encrypted, salted, and dispersed among many nodes on the STORJ network.

If data are not recoverable locally, then it is necessary to turn to the STORJ network. This means that investigators will need to interact with STORJ's API to recover data and metadata. A username and password are required to gain account access, and the acquisition of credentials may not always be trivial. Furthermore, even with credentials, a passphrase is required to download each file necessary for an investigation. The shards are salted with the passphrase, which makes recovery even more complicated: if all shards are recoverable, the passphrase is still required to decrypt the encrypted salted shards.

Another challenge investigators may face is identifying the location of file shards. The identifiers used to determine where shards are located are ambiguous and do not provide the necessary location information. Even if investigators could determine shard locations, warrants may be required for every possible location each shard is in—meaning that tens or even hundreds of warrants may have to be issued, depending on the size of the file and other factors.

This presents additional concerns because no longer are one person and one service provider involved in an investigation but many. Additionally, if a file has been marked for deletion by the user, a message is sent to the location of the shards for them to be erased. This makes being able to determine whether those shards existed impossible, given that there will be no metadata to indicate they did.

Many of these challenges may be considered unprecedented and are a testament to the complexities of data acquisition as storage technology advances. We are moving to an age where all data will be stored in the cloud, and, as we see with blockchain distributed cloud storage, data acquisition becomes incredibly difficult. The following sections present preliminary research on STORJ forensics, along with potential remedies to help investigators recover data from the STORJ network in a forensically sound and worthwhile manner.

Preliminary Research

We conducted preliminary forensic research on the blockchain-based storage service STORJ. This section does not present the full scope of the recovered artifacts but aims to show why more work needs to be conducted on the forensic analysis of blockchain storage services. During our initial analysis, we set out to obtain client- and server-side artifacts and attempted to recover files uploaded to our client machine by other STORJ users.

1. *Setup*: To allow other users on the STORJ network to use our machine's storage, we had to first download STORJ Share, an application designed for Windows that allows the user to allocate disk space to be used for renting. A Bitcoin address also had to be created to receive the STORJCoin (STORJ's digital currency). Once those two items were configured, the machine was left running for over a month, allowing STORJ users to upload their shards to our machine. During this process, we analyzed another STORJ application that allows users to interface with their accounts. The STORJ website does not let a user directly manipulate files, as commonly implemented in Google Drive, Dropbox, and the like. This required us to download several tools on Windows, i.e., Bash for Windows, npm, and Node.js. After this was completed, we could connect to our account on the STORJ network via private/public elliptic curve digital signature algorithm keys.
2. *Results*: With the setup completed, we set out to recover all relevant and interesting artifacts. First, we attempted to identify whether we could directly read any shards from the disk storage space we designated. The first file we came across was an .ldb file that contained the transactions between the farmer (client machine) and another STORJ user who wanted to upload files. Upon analysis, there were no data that pointed to identifying the type of file the shard could have come from. From the transactions, we could identify a hexadecimal contract number, farmer ID, signature, payment destination, and several other interesting items that could have had some significance in a forensic investigation.

On the other side of the analysis, we aimed to recover artifacts from the STORJ network side. We could upload files to the network after creating an account. The application used to upload the files was command-line based. With the files uploaded, we were able to recover only some metadata, i.e., the date/time the file was uploaded, name of the file, file size, and type of file. After uploading, downloading, and recovering the metadata, we deleted the file to see if it was at all possible to recover any metadata or retrieve the file. Our initial analysis indicated that there was no way to recover these data. However, given that this was a preliminary analysis of one of the few blockchain-based

storage platforms, more work needs to be performed to validate our initial findings.

Our primary results indicated that recovering artifacts is not as trivial as it may seem and that additional research is required to provide a fully comprehensive analysis of the artifacts that may be recovered from services like STORJ. There is still significant work that needs to be carried out, as discussed in the following section.

Future Directions

There are many challenges with blockchain-based distributed storage forensics, one of which is recovering files and metadata that can be useful in a prosecution. This challenge must be countered, because it is not guaranteed that such data are recoverable on a suspect's local storage. Given that STORJ is a new service, a clear investigative methodology is required to assist examiners in the forensic reconstruction of evidence without jeopardizing a case's outcome. This means that the construction of a forensically sound methodology for recovering evidence and artifacts from both the farmer and client on the STORJ network is an imperative area of research.

This leads us to the analysis of digital artifacts that have been created by STORJ. There is no work to date that has identified all of the artifacts produced by this service. Last, we propose an applied research direction that provides a tool for interfacing with STORJ's API and that can recover metadata and files that are potentially crucial for a case.

Methodology

The first proposal to remedy the challenges identified in blockchain-based storage forensics is to develop an extensible, forensically sound, and peer-reviewed methodology that will assist investigators in recovering evidence. The methodology should employ a clear step-by-step process that can provide investigators with hints as to where artifacts may be found and what to keep in mind when investigating a case involving systems like STORJ. This will ensure that there are no steps missed and that, when an investigator presents evidence in a court of law, he or she can confidently claim that the methodology used has been studied, published, and found to be forensically reliable. In this way, no challenges can be made against the investigation regarding the integrity of the methodology used to recover incriminating information.

Artifacts

Identifying artifacts created by any service on computers is crucial to an investigation. Artifacts illustrate that something has been conducted on a machine or may indicate something suspicious that occurred.[19] Because STORJ and the artifacts it produces have not yet been studied, it is crucial to identify them. These artifacts can help investigators confirm that STORJ was indeed used on the machine, the nature of its use (e.g., to rent out storage space or upload data), and whether further analysis of the machine is required before ruling it out as insignificant for a case. When one can determine how STORJ was used, the data necessary to help access data in the STORJ network (identifying a username, a password, the private key, and/or potential passphrases) may then be collected. With the necessary data collected, a tool can be used to find additional evidence in the STORJ account, a procedure discussed in the following section.

Tool

The tool we propose constructing requires credentials, which would allow investigators to connect to and interact with the STORJ API. From here, investigators can have the option to download files, recover metadata, and discover several other pieces of evidence that can help them build their case. This tool would need to ensure that the data are authentic and should allow the recovery to take place in a forensically sound manner. There should be nothing that could potentially alter the data being recovered (in which case the information could not be used in court). Ideally, this tool would also attempt to retrieve data that could be used to locate essential evidence on the local machine, because this would help automate and speed up the digital forensic examination process.

Computer storage has progressed significantly over the past century; with it, many challenges have arisen in regard to digital forensics. From local storage devices (e.g., HDDs and memory) to cloud-based stor- age services (e.g., Google Drive and Dropbox), investigators have had to conquer several challenges related to evidence recovery. Many of these have been overcome with ingenuity and tools to help in the process of recovering incriminating data. However, we have discussed an upcoming hurdle in digital forensics—blockchain-based cloud storage.

In this article, we explored the plethora of challenges that investigators must face related to this new cloud storage technology, such as distribution of shards, default encryption, determining the location of shards, recovering files with user credentials, and so on. These challenges must be met with a forensically sound methodology, identification of artifacts, and a tool to assist investigators in retrieving artifacts and telltale evidence. It is necessary to bring to light the potential this

technology has and what can be done to assist in digital forensic investigations.

References

1. M. Taylor, J. Haggerty, D. Gresty, P. Almond, and T. Berry, "Forensic investigation of social networking applications," *Netw. Security*, no. 11, pp. 9–16, Nov. 2014.
2. H. Chung, J. Park, S. Lee, and C. Kang, "Digital forensic investigation of cloud storage services," *Digital Investigation*, vol. 9, no. 2, pp. 81–95, 2012.
3. G. Grispos, T. Storer, and W. B. Glisson, "Calm before the storm: The challenges of cloud computing in digital forensics," *Int. J. Digital Crime Forensics*, vol. 4, no. 2, pp. 28–48, 2012.
4. C. Hooper, B. Martini, and K.-K. R. Choo, "Cloud computing and its implications for cybercrime investigations in Australia," *Comput. Law Security Rev.*, vol. 29, no. 2, pp. 152–163, 2013.
5. National Institute of Standards and Technology, "NIST cloud computing forensic science challenges," June 2014. [Online]. Available: http://csrc.nist.gov/publications/PubsDrafts.html#NIST-IR-8006
6. D. Quick and K.-K. R. Choo, "Google Drive: Forensic analysis of data remnants," *J. Netw. Comput. Appl.*, vol. 40, pp. 179–193, Apr. 2014.
7. B. Martini and K.-K R. Choo, "Cloud forensic technical challenges and solutions: A snapshot," *IEEE Cloud Comput.*, vol. 1, no. 4, pp. 20–25, 2014.
8. J. Dykstra and A. T. Sherman, "Understanding issues in cloud forensics: Two hypothetical case studies," 2011. [Online]. Available: https://commons.erau.edu/adfsl/2011/wednesday/10/
9. D. Neilson, S. Hara, and I. Mitchell, "Bitcoin forensics: A tutorial," 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-51064-4_2
10. M. Fukumitsu, S. Hasegawa, J. Iwazaki, M. Sakai, and D. Takahashi, "A proposal of a secure P2P-type storage scheme by using the secret sharing and the blockchain," in *Proc. IEEE 31st Int. Conf. Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan, 2017, pp. 803–810.
11. L. D. Jones, "Examining the forensic artifacts produced by use of bitcoin currency," Utica College, Utica, NY, 2014.
12. B. Martini and K.-K. R. Choo, "Cloud storage forensics: OwnCloud as a case study," *Digital Investigation*, vol. 10, no. 4, pp. 287–299, 2013.
13. V. Roussev, I. Ahmed, A. Barreto, S. McCulley, and V. Shanmughan, "Cloud forensics: Tool development studies & future outlook," *Digital Investigation*, vol. 18, pp. 79–95, Sept. 2016.
14. B. Martini and K.-K. R. Choo, "Distributed filesystem forensics: XtreemFS as a case study," *Digital Investigation*, vol. 11, no. 4, pp. 295–313, 2014.
15. M. Liberatore, R. Erdely, T. Kerle, B. N. Levine, and C. Shields, "Forensic investigation of peer-to-peer file sharing networks," *Digital Investigation*, vol. 7, supplement, pp. S95–S103, 2010.
16. S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer-to-peer cloud storage network," Storj, Atlanta, GA, White Paper, Rep. v1, 2014.
17. M. D. Doran, *A Forensic Look at Bitcoin Cryptocurrency*. Utica, NY: Utica College, 2014.
18. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
19. V. S. Harichandran, D. Walnycky, I. Baggili, and F. Breitinger, "CuFA: A more formal definition for digital forensic artifacts," *Digital Investigation*, vol. 18, supplement, pp. S125–S137, 2016.
20. Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Inform.*, vol. 8, no. 2, 2017. doi: 10.3390/info8020044.
21. R. C. Merkle, "Protocols for public key cryptosystems," in *Proc. 1980 IEEE Symp. Security Privacy*, Oakland, CA, 1980, p. 122.

Joseph Ricci is a consultant at Lodestone Security. He performs penetration testing and vulnerability assessments. Ricci received his B.Sc. in cybersystems from the University of New Haven, Connecticut, and he was a researcher at the University of New Haven's Cyber Forensics Research and Education Group. He was part of a team that placed third out of 184 in the international Black T-Shirt Cyber Forensics Challenge. Ricci coauthored a book chapter on smartwatch forensics and security in *Managing Security Issues and the Hidden Dangers of Wearable Technologies* (IGI Global, 2016). Contact him at jricc3@unh.newhaven.edu.

Ibrahim Baggili is the founder and codirector of the University of New Haven's Cyber Forensics Research and Education Group and is the Elder Family Endowed Chair of Computer Science at the university. Baggili received his Ph.D. in computer and information technology specializing in cybersecurity at CERIAS, Purdue University. He has published extensively in this field and has been awarded a number of federal grants. Contact him at ibaggili@newhaven.edu.

Frank Breitinger is an assistant professor of computer science at the University of New Haven and is codirector of the University of New Haven's Cyber Forensics Research and Education Group. Breitinger received his Ph.D. in computer science from the Technical University Darmstadt, Germany. Contact him at fbreitinger@newhaven.edu.