Degree Project

Bachelor's

An Analysis of Using Blockchains for Processing and Storing Digital Evidence

Author: Tobias Svenblad
Supervisor: Hans Jones
Examiner: Mark Dougherty
Subject/main field of study: Digital Forensics and Internet Security
Course code: DT2020
Credits: 15 hp
Date of examination: 29th of May 2018

At Dalarna University it is possible to publish the student thesis in full text in DiVA. The publishing is open access, which means the work will be freely accessible to read and download on the internet. This will significantly increase the dissemination and visibility of the student thesis.

Open access is becoming the standard route for spreading scientific and academic information on the internet. Dalarna University recommends that both researchers as well as students publish their work open access.

I give my/we give our consent for full text publishing (freely accessible on the internet, open access):

Yes ☒
No ☐
Abstract:

A review of digital forensics today shows that it could be exposed to threats jeopardizing the digital evidence integrity. There are several techniques to countermeasure this risk, one of which is the method that involves the use of blockchains. Blockchains use an advanced system to keep the data within it persistent and transparent, which makes it a natural candidate for everything integrity-sensitive. Several blockchain techniques and infrastructures have been described in this study, based on previous studies and other literature work. Interviews and experiments made a comparison between traditional digital forensic methodologies versus blockchains possible in later chapters. The results showed that blockchains could be the answer to securing digital evidence integrity. However, there is still a lot more work to be done before blockchains are ready to be implemented in production systems. The results of the blockchain analysis are presented such that they can be used as an aid to further research, either theoretically or practically, digital evidence blockchains.

Keywords: Blockchain, Digital Forensic Blockchains, Digital Evidence
# Table of Contents

Table of Contents .................................................................................................................. 3  
Table of Figures ...................................................................................................................... 5 
Terminology .............................................................................................................................. 6 
1. Introduction ......................................................................................................................... 7  
  1.1. Background ..................................................................................................................... 7  
  1.2. Purpose .......................................................................................................................... 7  
     1.2.1. Goals ....................................................................................................................... 8  
  1.3. Research methods ......................................................................................................... 8  
  1.4. Scope ............................................................................................................................. 8  
  1.5. Previous studies ........................................................................................................... 8  
2. Blockchain Introduction ....................................................................................................... 9  
  2.1. The Problem Today ...................................................................................................... 9  
  2.2. User Scenarios ............................................................................................................. 12  
  2.3. Blockchain Technology ............................................................................................... 13  
     2.3.1. History of Blockchains ......................................................................................... 13  
     2.3.2. How Blockchains Operate ................................................................................... 13  
     2.3.3. Discovery of Nodes ............................................................................................... 16  
     2.3.4. Types of Blockchains ......................................................................................... 16  
3. Suggested Model Infrastructures ....................................................................................... 18  
  3.1. Consortium Blockchain - Police Departments ............................................................... 18  
  3.2. Public Blockchain - Chain of Custody .......................................................................... 19  
  3.3. Private Blockchain - On Premise Read Access ............................................................... 21  
4. Processing Digital Evidence with Blockchain Technology ................................................. 22  
  4.1. Block Content and Block Size ..................................................................................... 22  
     4.1.1. Whole Evidence Files .......................................................................................... 22  
     4.1.2. Image File Hashes ............................................................................................... 23  
  4.2. Practical Implementation .............................................................................................. 24  
  4.3. Suggested Application ................................................................................................. 26  
5. Comparisons ......................................................................................................................... 29  
  5.1. Security ......................................................................................................................... 29  
  5.2. Performance ................................................................................................................ 31  
  5.3. Cost ............................................................................................................................... 32  
  5.4. Scalability .................................................................................................................... 33  
6. Discussion ........................................................................................................................... 33  
7. Conclusion .......................................................................................................................... 35  
8. Further Research ................................................................................................................. 35  
References ............................................................................................................................... 37
Appendixes ................................................................................................................................................. 41
Appendix A .................................................................................................................................................. 41
  Blockchain.java ....................................................................................................................................... 41
  Block.java .............................................................................................................................................. 42
  StringUtil.java ..................................................................................................................................... 43
Table of Figures

Figure 1: A visual representation of how hash fingerprint function works (Stolfi, 2008). 10
Figure 2: A forensic disk controller used to ensure the disk is only read from and not written to (ErrantX, 2010). 11
Figure 3: A visualization of how a block looks like (Decuyper, 2017). 14
Figure 4: An example of how hashes of previous blocks are used to form the chain (Decuyper, 2017). 15
Figure 5: Changing some data within a block causes all the following blocks to become invalid (Decuyper, 2017). 15
Figure 6: A visual representation of how public, consortium and private blockchains could look like. The orange dots represent nodes with read/write permissions where the white dots represent nodes with only read permission. 17
Figure 7: A consortium blockchain; could be used by Police Departments nationally or internationally. 19
Figure 8: A public blockchain; could be used as a chain of custody model. 20
Figure 9: A private blockchain; useful for internal purposes. 21
Figure 10: Some forensic toolkits like AccessData’s FTK will display the hash of the digital evidence image (Austin, 2013). 24
Figure 11: A UML diagram of the blockchain project example. 25
Figure 12: The result output when running the practical example. 26
Figure 13: An example of how the application could look like. 28
Terminology

- **Blockchain**: A system which keeps track of transactions written to a digital ledger stored on several computers linked in a peer-to-peer network.

- **Block**: The data sent to the blockchain. A block usually contains the data, the hash of the block, the hash of the previous block, and a timestamp.

- **Peer-to-peer**: A network of computers participating in sharing tasks or workloads distributed from an application.

- **Node**: A computer or party participating in the peer-to-peer network.

- **Ledger**: A physical book or digital file for keeping notes, or entries, typically used for accounting and transaction records.

- **Cryptographic Protocol**: Performs a security-related function and applies cryptographic methods to encrypt data transmitted between a network.

- **IRC**: Internet Relay Chat; an application communication protocol that allows clients and servers to communicate via text form.

- **DNS**: Domain Name System; a hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It translates IP addresses to domain names.
1. Introduction

1.1. Background

We live in a constantly digitizing society with more and more smartphones and computers entering the market every day. This rapid increase of digital devices certainly has its benefits; we see an increased rate of productivity, ease of accessibility, a cost-effective management accompanied with great documentation (Aptara Corp, 2017). In retrospect it comes with a drawback; increased digital crime rates (Travis, 2015).

While we probably won’t be able to completely stop digital crimes from happening, there are improvements we can make in the way digital evidence is processed and stored. Today, most digital forensics use a more traditional style when it comes to investigating digital crimes. The evidence is seized and plugged into a special device controller connected to a special computer where the investigator will create a forensic copy of the evidence (a forensic copy is an exact replica taken, sector by sector, from a hard disk or a flash drive). From there, the digital forensic will investigate the data to see what can be found against said copy. Lastly, a report is created and send together with the evidence to the police chief or any other higher authority.

In the perfect world, after taking the forensic copy of the evidence, the digital security and evidence integrity would never be compromised. However, what if someone alters the evidence once the device is taken into custody? Or even worse, cracks into the computer system used by the investigator to make changes to the evidence copy or the metadata of the evidence files? A possible solution to this problem is blockchain technology.

The blockchain technology is difficult to define, but generally it is often considered to be a persistent, decentralized, transparent, append-only ledger (Jeffries, 2018). This ledger is what makes the blockchain so special. It allows new data to be added to the blockchain, but it doesn’t easily allow changes to previous data within. The ledger is distributed across several parties, or nodes, in a network that is used to sign and verify each new data added to the ledger. On top of this, every time new data is added or changed, the nodes in the network will need to reach a consensus as to include it in the blockchain. This creates a great way of verifying that the data is correct and hasn’t been tampered with. There are existing systems today that work great for this purpose as well, but considering the high number of cybercrimes happening today, the integrity of evidences might be jeopardized. Blockchains might hold the answer to that. Blockchains are astoundingly complex but could provide the needed solution to process and preserve digital evidence without adventuring integrity and accessibility.

1.2. Purpose

The purpose of this study is to investigate the possibility for police authorities to use blockchain technology when processing and storing digital evidence.
1.2.1. Goals
This study has two main goals:

1. This study aims to analyze how new and existing blockchain solutions could be used to process and store digital evidence in its many forms.

2. This study will also analyze the potential benefits or drawbacks of digital forensic blockchains versus traditional forensic methodologies.

1.3. Research methods
In this study, a mix of different research methods were used to reach the end conclusion. Interviews were carried out with a digital crime investigator from Mariehamn, Finland to paint up a picture of how the current system is functioning and what it lacks (different drawbacks). Comparisons were then prepared between the already existing system and the proposal discussed in this paper.

An experiment in the form of a practical implementation is also done in chapter 4.2. Practical Implementation. Other than that, the study is based off case studies to receive the needed in-depth knowledge of how blockchains work and what technologies they use. Recent documents and research papers touching the area has also been analyzed and referenced in this study as part of the infrastructure design suggestions discussed in later chapters.

1.4. Scope
This study will not include any advanced technological details to explain how blockchains operate or how to build large-scale versions practically. A simple practical example is given, but it is not to be considered a ready-for-production blockchain. The scope is further narrowed by only applying the theory on digital evidence, not physical evidence.

While this study could be applied to a lot of institutions carrying out similar activities as digital forensics in the police force, the focus lies mainly on police authorities.

1.5. Previous studies
There exists a handful of studies on blockchains used for criminal evidence storing and processing, but most of them are not very broad and doesn’t offer any comparisons or practical implementations around the subject, but they do offer some insight in how the digital forensic process might benefit if any such technology were to be implemented.

One of the more head-on papers of the topic is *Blockchain of Evidence*, written by David Salgado in 2016 (Salgado, 2016). Salgado’s paper is based on the idea of digital images and videos one might use as criminal evidence in court, where the prosecutor has one version of the image and the defense has another version. The images are both equally authenticate and both sides claim their version is the original one. To solve the dispute, an expert will walk the jury through the process of checking the integrity of the digital images against its digital fingerprint, stored
on the blockchain when the image was accepted into evidence. This way, the argument can be settled in an easy and righteous manner.

Another study, performed by Lone & Mir, explains that blockchains may be used in chains-of-custody, which might be useful when sharing evidence between all parties in a court hearing. (Lone & Mir, 2017). A study by Sugrue, titled *Digital Video Chain of Evidence in the 21st Century*, explores the idea of using blockchains to store digital evidence from video cameras directly. “Video evidence,” Sugrue says, “whether from CCTV, covert surveillance cameras, mobile or body-worn cameras, is important in the majority of criminal investigations.” (Sugrue, 2017).

### 2. Blockchain Introduction

This chapter and the underlying chapters will explain the issues digital evidence might face today regarding its integrity. The history of blockchains and how they work theoretically will be described later.

#### 2.1. The Problem Today

Before the computer became mainstream, photos, videos and documents only existed in their physical form. If any such item is used as criminal evidence today, they are often stored and locked away in a special evidence room to where only authorized people have access to. With the widespread usage of digital devices however, photos, videos and text documents rarely exist in their physical form anymore; anyone who has snapped a picture of their holiday trip to Bermuda are more likely do so from their cell phone rather than taking out their old polaroid camera. These digital files and any other file that may be saved on a computer, are called digital assets.

These digital assets are fundamentally just collections of ones and zeros. These ones and zeros are what determines the uniqueness of that asset as it can be calculated in a so called ‘hash fingerprint function’. The hash function acts like a unique fingerprint to any digital asset of any size or type, and the chances of two different assets ending up having the same hash value are negligibly small; it took a team of researchers about 6,610 years of processor time to find the same hash value of two different digital assets (Leyden, Claburn, & Williams, 2017). This practically means that there is no digital asset that has the same hash fingerprint.

It is often considered crucial that the hash value doesn’t change once the evidence has been seized and processed. If anything changes in the file (e.g. if a lower-case letter would change to an upper-case), the hash fingerprint will change drastically. The file can therefore no longer be considered to be the same piece of evidence (example at Figure 1) and will not hold in court (Lorraine v. Markel American Ins. Co., 2007).
To countermeasure the problem of hashes changing, there are methods the investigator can apply. Forensic disk controllers (Figure 2) will make sure a hard disk drive can only be read from but not written to, to make sure no information is altered accidentally by the investigator, thus not changing the hash fingerprint. In addition to this, some digital forensic tools used to examine the material found can also offer built-in functionality to take a forensic copy (often known as an image) of the disk drive and save it in another location to make sure the investigation is done against the copy instead. This practice makes sure the original copy remains intact and will further increase the security and the integrity of the digital evidence.
Once the copy of the hard disk has been made and all the evidence has been found and processed, the digital forensic examiner will often write a report stating what he found and send it off to the police chief or any other higher authority. However, once the evidence has been handled on the investigator’s computer, there’s no telling what illicit activities are made against it.

As Salgado wrote in his paper, *Blockchain of Evidence*, 2016, “Digital assets can be copied at will, leaving no trace and losing no fidelity in the process. In the context of criminal evidence, this makes it hard to restrict access to a digital asset.” This means that if the access to the internal infrastructure of the Police Department isn’t restricted, system administrators, colleagues and cloud service providers could easily open and enter the copied evidence and could even change it, unintentionally or on purpose, rendering the evidence found unusable or erroneous in court.

Salgado also mentions in his paper that you can lock a copy of the evidence in a safe by saving it on a hard disk, but it may be difficult to know how many copies of that evidence are flying around digitally inside or even outside of your organization, and who has access to them. **Error! Bookmark not defined.** If an investigator checks the hash fingerprint on an evidence file, examines it, creates a report out of the evidence found and lock it up in a safe once done, will he really remember the hash fingerprint? Will he remember exactly what he found on that evidence? Probably not, especially if the investigator has a lot to do daily. If someone then enters the investigators computer and changes some evidence
material to their or the perpetrators benefit, will the investigator notice the changes?

The blockchain technology will be discussed in the following chapters, but first, let’s look at a couple of user scenarios to where blockchains might be especially useful.

### 2.2. User Scenarios

Arguably, if the risk of digital evidence tampering is put into perspective, it may seem a little farfetched. Who would manipulate the digital evidence acquired from a computer or a cell phone? What good would it do? Well, it has been shown time and time again that even though police officers and special agents are supposed to be lawful workers, there will always be someone corrupt who might not work against best practices (Shin, 2014) (Dwyer, 2011). The gist of it all; trust no one when processing and storing digital evidence.

Consider the following hypothetical situations, created by Salgado (Salgado, 2016):

1. “A tourist’s photograph is a key piece of evidence in a criminal case. In the background of the image, the suspect can be seen holding a gun. The defense presents the same picture, but in this version the suspect is holding a coffee cup. Both the prosecution and the defense argue that theirs is the genuine image, and the other version has been doctored. The jury does not know who to believe.”

2. “A defendant claims he was beaten by the police, during his arrest. Somehow, a video clip from the body camera of one of the arresting officers is leaked. The clip supports the defendant’s claim. The police force in question later issues a statement to the effect that the clip has been edited, and vital context removed. They release what they say is the original footage. Arguments rage on the internet over which version of the incident is true.”

Now, imagine the same two examples in a world where digital evidence is authenticated on a distributed ledger (blockchain);

1. “One side in the case brings in an expert who walks the jury through the process of checking the integrity of the digital image against its digital fingerprint, stored on the distributed ledger when the image was accepted into evidence. He explains why this is important, then repeats the process using the other version of the image, showing proof that it has been altered.”

2. “The ‘official’ version of the video clip is released with instructions enabling anyone in the world to use the public blockchain to verify its integrity for themselves. Most people don’t understand how this works, but enough of them (including most of the journalists) have friends who do
understand it that the allegations of misconduct are dismissed by all but the most hardcore conspiracy theorists.”

2.3. Blockchain Technology
The following chapters will focus on the history of blockchains and how they work in theory.

2.3.1. History of Blockchains
The earliest work that resembles the blockchains created today, was carried out by Stuart Haber and W. Scott Stornetta in 1991 (Haber & Stornetta, 1991). Their white paper talks about the technology in a way of verifying time-stamps on documents and the study concluded that: “[document] time-stamping could be extended to enhance the authenticity of documents for which the time of creation itself is not the critical issue.” It was improved further one year later in 1992, when the use of Merkle trees was added to the design which made it more efficient by allowing several documents to be collected into one block, which would become an important stepping stone 16 years later, when the first modern blockchain technology was created (Bayer, Haber, & Stornetta, 1992).

The modern blockchain was invented by an unknown individual going under the pseudonym Satoshi Nakamoto in 2008. The goal of the blockchain, according to Nakamoto, was to host a public transaction ledger on the blockchain for the cryptocurrency Bitcoin (Nakamoto, 2008). The goal of the entire project, however, was to create a decentralized digital currency which would solve the double spending problem. Double spending is a flaw in the digital cash scheme in which a party can spend the same digital token more than once, leading to inflation by creating new fraudulent currency that did not previously exist; hence, a third party is needed to regulate the double spending, which is not needed when using Bitcoin (Chohan, 2017).

After the launch of the Bitcoin and the blockchain technology it was based on, the cryptocurrency saw an enormous increase of users during 2014 through 2017; the blockchain size grew from around 20 GB in 2014 to nearly 100 GB in 2017, which had only previously been miniscule (Bitcoin.com, u.d.). During this steadily growing success of Bitcoin, many other applications followed in the same steps and made their debut using blockchain technology for a wide range of purposes. The smart contracting blockchain Ethereum is probably one of the most well-known example (Ethereum Community, 2016). Smart contacting is, according to CoinTelegraph, “a special protocol intended to contribute, verify or implement the negotiation or performance of the contract. Smart contracts allow to perform credible transactions without third parties” (Tar, 2017). Blockchains are even used to securing legal processes for real estate transfers (Ubiquity, 2015).

2.3.2. How Blockchains Operate
The blockchain technology, as mentioned in the Introduction, is difficult to define, but often it is a persistent, decentralized, transparent, append-only ledger (Jeffries, 2018). The ledger is simply another word for the blockchain itself and it is distributed across multiple computers, or nodes, working independently of one another; however, a consensus between the nodes must be met before data can be
added or altered in the blockchain. It is often easy to add new data to the blockchain, but it doesn’t easily allow you to change previous data within it, which is what makes blockchains such an interesting technology for sensitive data.

The below paragraphs will explain how blockchains work theoretically, and is not tied to any specific blockchain existing today (like Bitcoin or Ethereum).

The blocks that make up the blockchain contains several pieces of important information, namely some type of data, the hash of the current block, the hash of the previous block, and occasionally, a timestamp (Figure 3). The data stored inside the block depends heavily on the type of blockchain. For instance, the Bitcoin blockchain stores all transaction details (sender, receiver, and number of Bitcoins) as its data, but the data could be anything; medical records, tax information, contract details, or digital evidence.

![Image: Visualization of how a block looks like](image)

**Figure 3:** A visualization of how a block looks like (Decuyper, 2017).

The hash of the current block can be compared to that of a fingerprint; it’s always unique and will be calculated when the block is being created. If a change would be made to the block after the hash is calculated, the whole hash will change for that block. It is, in other words, a very good way to detect changes in the data once the block has been added to the blockchain, because once the data changes, it’s no longer the same block, even if you would change just the smallest bit of data, like adding a dot to a sentence or changing one single pixel in an image.

The same kind of logic is used with the hash of the previous block. All blocks in a blockchain will contain the hashes of the block itself plus the hash of its immediate preceding block. Since the first block in the blockchain cannot contain any hashes from previous blocks, the value is usually only zeros and is called the Genesis Block. The hash of the Genesis Block will however be contained in the following
block in the chain. That block, in turn, will calculate its own hash, which the third block will read to make out its own hash, and so on (Figure 4).

If the hash in a block changes, the chain will be broken and all succeeding blocks will have to recalculate their hashes as well (Figure 5). Since each block’s header contains a construct, or Merkle root, changing the data in block 2 will make the Merkle root in the header of block 3 to become invalid. If the header of block 3 changes, so will the header of block 4, and so on and so forth. This creates an effective chain of blocks which is the foundation of which the blockchain technology relies on.

**Figure 4:** An example of how hashes of previous blocks are used to form the chain (Decuyper, 2017).

**Figure 5:** Changing some data within a block causes all the following blocks to become invalid (Decuyper, 2017).
2.3.3. Discovery of Nodes

If a new node joins the peer-to-peer blockchain network - how will that node locate all the other nodes? A blockchain needs a way of locating all nodes to fully achieve the proper consensus protocol and become a truly decentralized and immutable ledger.

There are many ways a node can connect to the rest of the blockchain. Unfortunately, many blockchains utilize their own ways of doing this. For example, the Bitcoin blockchain would historically use IRC seeding. The node would randomly join an IRC channel from a hard-coded IP address in the blockchain application. Then the node would issue a WHO command and read the IP addresses from other nodes appearing in the channel. Unless the node was turned off, it would keep asking the channel in a loop indefinitely to receive any updates of new nodes joining the network.

Another, less resource intensive way, is that the blockchain application can hard code certain IP addresses that lead to powerful and consistent nodes, many of which are never turned offline or have several replicas of themselves in several parts of the world in case the original accidentally goes offline. These powerful nodes can then send a list of all nodes in the network, since they are all connected to that powerful node in the first place.

Today however, the most common way for new nodes to find other nodes in the network is by connecting to a specific hard-coded DNS address in the application. Sending a DNS request to the hard-coded DNS server will result in the sender to receive a list of nodes so that it can begin discovering the network. Sometimes, a blockchain can utilize several of these methods in case one method fails. That way, you always have a backup way to connect to the other nodes (Bitcoin.it, 2017). Of course, DNS isn’t always bulletproof, and it is discussed further in chapter 5.1. Security.

2.3.4. Types of Blockchains

To understand what type of blockchain would be suitable for evidence materials, this chapter will explain all existing types of blockchains: public blockchains, consortium blockchains and private blockchains (Figure 6).
The public blockchain was created together with the Bitcoin cryptocurrency. It is, as the name suggests, publicly available to anyone with an Internet connection who wishes to use it. With absolutely no access restrictions, anyone can read and add data to the blockchain and anyone can become a node in the network to help validate new blocks and participate in the consensus protocol. A public blockchain is therefore considered to be a fully decentralized blockchain. Some examples of public blockchains are the cryptocurrency Bitcoin and the smart contract platform Ethereum (Goel, 2015).

The consortium blockchain is not as open and accessible as its public counterpart. You can think of the consortium blockchain as a smaller variant than the public blockchain; it is controlled by a pre-selected set of nodes, and is usually considered a partially decentralized blockchain. The pre-selected set of nodes is usually like-minded companies and institutions, like banks, hospitals, or police departments. An example of such a blockchain is the Energy Web Foundation; a blockchain aimed to accelerate the use of blockchain technology across the energy sector (Energy Web Foundation, u.d.). Most of the time, the same type of consensus protocol is still needed in consortium blockchains, so if ten banks are running the same consortium blockchain, at least six of them would have to agree on any change made to that blockchain (Buterin, 2015).

The private blockchain is often owned and used by a single organization where write permissions are kept centralized to that organization. The read permissions on the other hand, may be public or restricted to only the organization itself for internal use. The consensus protocol doesn’t exist the same way as in the public and consortium blockchain, so it doesn’t need to worry about the distribution of
nodes, since the organization owns 100% of them. Private blockchains could be used for things like database management or internal auditing (O'Connell, 2016).

3. Suggested Model Infrastructures

There is no real definitive answer to how a blockchain infrastructure should look like. Since blockchains are very flexible, there’s no one-answer-fits-all best practice to follow, as is usually the way with other technologies. However, there are a couple of ways to design what type of blockchain that should be built and what result desired to achieve. The suggested infrastructures in this chapter aims to give a basic overlay of how one could spread out the nodes and in which way these could work together.

3.1. Consortium Blockchain - Police Departments

The first suggestion as a hypothetical infrastructure design is aimed to only be used between Police Departments across a country, or even cross-border, depending on national law. Because the blockchain will be shared only between like-minded organizations, the type of blockchain will be of a consortium nature (Figure 7).

The blockchain could be based on individual cases that would have their digital material uploaded to the blockchain, which would automatically be sent to all other nodes in the network (the different Police Departments). Not only does this provide a great way of sharing case details in a quick matter if, e.g. the suspect has fled from one country to another or if the suspect has committed crimes across multiple municipalities touching several areas of which different Police Departments act, but it would also provide an automatic and accessible way of timelining the investigation process. For instance, if the perpetrator murders a person in City A on the 1st of January and then murders again in City B on the 5th of January, you can easily begin to map out where and when the murders took place by just looking at the audit trail of the blockchain.
Figure 7: A consortium blockchain; could be used by Police Departments nationally or internationally.

A network of this type could act as a criminal information and intelligence database, which could be used to combat many areas of specific crimes like sexual exploitation of children, much like the already existing Europol Information System (EIS), launched in 2005 (EUROPOL, 2015). For example, if an investigator finds an illegal picture on a suspect’s computer, he could upload it to the EIS to see if that picture has been encountered before on someone else's computer across the globe. This gives the investigator an insight into how involved that pedophile is and if he has committed the sexual acts himself, or if he has only downloaded the pictures from somewhere.

A possible downside of a design like this is that it’s not always relevant for every police department in the country to receive information about a local computer theft happening in City A. This could cause unnecessary congestion in the network and the blockchain could grow quite large if every less serious case were to be shared across the blockchain. This is further discussed in chapter 4.1.1. Whole Evidence Files.

3.2. Public Blockchain - Chain of Custody
The second infrastructure suggestion is the Chain of Custody model which would be beneficial if the need for a precise audit trail of evidence; from first responder investigators to higher authorities (Figure 8).

This model is based on A. H. Lone et al.’s paper Forensic-chain: Ethereum Blockchain Based Digital Forensics Chain of Custody, 2017. A. H. Lone explains that the chain of custody can be defined as the chronological history of processing and storing digital evidence. When the digital evidence passes through the
different levels of hierarchy (from first responder to higher authorities), there is always higher degree of integrity violation and repudiation (Lone & Mir, 2017). The system is based on Ethereum, a smart contract blockchain in which anyone can build their own application fueled by Ethereum’s own cryptocurrency Ether. The blockchain discussed in the paper uses the smart contracts to allow evidence tracking and will help in detection when copies of evidence are being made, so that only authorized copies or other types of housekeeping or record keeping would be specially entered in the chain of custody of the blockchain. Since the blockchain makes an almost permanent record of the chain of custody, with the evidence digitally preserved, no evidence is thrown out.

The usage of Ethereum isn’t necessarily needed in the execution of a design such as this one; it could be run under its own blockchain as well. In the model put forward a public party is also involved, depending on which route the legal system wants to take. There could be, for instance, an application anyone could download to communicate with the blockchain but only be able to see, not write, to the blockchain. The progress of different cases and in what status they are could be made public, but not the actual evidence files. If someone committed a robbery or a murder, the public would be able to read about it in the blockchain; who was the victim, when and where it happened and when the police will begin their preliminary investigation.

The defense, court and prosecutor being able to read and write to the blockchain would have its benefit in the form of information contribution. The provenance of

*Figure 8: A public blockchain; could be used as a chain of custody model.*
any event or action can be traced back to where it originally entered the process in question.

This design has its obvious drawbacks however; apart from the cost increase in hardware and electricity, regulations such as the General Data Protection Regulation (GDPR) (EUGDPR, 2017) and the Directive on Security of Network and Information Systems (the NIS Directive) (European Commission, 2016) also makes it more difficult to store and share information to the public. It would take a lot of time to build an infrastructure with these regulations in mind, and even then, not all information about the case could be made public.

3.3. Private Blockchain - On Premise Read Access

This model is based on an on-premise blockchain where employees have read access but the digital forensics, who have read and write access. This type of blockchain suits the needs for any organization looking to keep an internal blockchain infrastructure in-house. The blockchain won’t be decentralized though and won’t solve the issue with corrupt police workers. In fact, the private blockchain isn’t much more than an internal database - used for forensic purposes (Figure 9).

![Diagram of private blockchain](image)

Figure 9: A private blockchain; useful for internal purposes.

Private blockchains have contradicting opinions however; one side says it shows great potential while the other side mean it’s nothing more than a cumbersome database (Narayanan, 2015). Max Kordek, CEO of Lisk, says that “A private blockchain is a great first step towards a more cryptographic future. The biggest advantages of private blockchains in comparison to centralized databases are the cryptographic auditing and known identities.” He continues to say that a private
blockchain is also cheaper, faster and could offer better privacy than a public blockchain (O'Connell, 2016).

Nevertheless, the private blockchain could offer the needed infrastructure to provide an immutable ledger of digital evidence. The issue is that many solutions already offer this kind of protection today; it’s not rare to implement change-alerts or audit tracking in a database.

4. Processing Digital Evidence with Blockchain Technology
The following chapters will analyze what type of block content would be suitable for the digital forensic blockchain. Later, a practical example will be shown as well as a prototype design for the blockchain application.

4.1. Block Content and Block Size
Before the blockchain is built, there must be a decision call as to what the actual block will look like and contain. Should it contain the whole evidence files, or only the hash value of the image files? This chapter will discuss both positives and negatives as well as analyze the different block sizes that could go with them.

Before diving into the aforementioned topics however, a discussion about block times and proof-of-work is needed to fully understand block sizes. The block time is the average time it takes for the network to generate a block once the previous block is full and has been appended to the blockchain; this can range from every five seconds to every ten minutes or more. Bitcoin, for instance, has a block time of 10 minutes, while Ethereum has a block time of between 10 and 19 seconds (Siriwardena, 2017).

The block time is utilized by the blockchain’s proof-of-work, which is a technique put in place so that there is always that short delay when adding or changing blocks, so that no one can change the blocks with rapid succession without anyone noticing.

4.1.1. Whole Evidence Files
If the block contained the evidence files themselves (i.e. a live preview of the files, so that you could open the case inside the blockchain to examine every individual pieces of evidence), the blockchain would be extremely complex. It would require a lot of effort to build, but could be doable. The blockchain would however grow very large very quickly. A homemade movie can easily take up several megabytes or even gigabytes in size, depending on the length and format of the movie. Images are no exception - they too can reach up to several megabytes in size. This would mean that the block time would have to increase as well to not congest the network, depending on what blockchain architecture is used.

This means that for every piece of evidence the investigator is examining, there will always be an alternating amount of data to be processed and sent to the blockchain. For this very reason, traditional block sizes wouldn’t be practical. Bitcoin, for instance, utilize the traditional block size and has an upper limit block
size of 1 MB. If the digital forensic blockchain were to utilize a block size of 1 MB as well, not only would it be too small in some cases, but the blockchain would downright ignore any file being uploaded to the blockchain that is larger than the upper size limit.

Ethereum, on the other hand, has no direct upper limit of block size, which is more attractive in a digital forensic point of view. Ethereum uses something called gas limit, which is a measurement of computational power, and essentially, a measurement of how big the block is going to be. This gas limit needs to be agreed upon before the actual computation takes place. The block gas limit is currently 3,141,592 gas which can be spent maximum per block; so, while there is a hard limit of how much gas you can spend per block, theoretically the block could grow even further (White Paper, u.d.).

While Ethereum might seem like a good choice when it comes to block size technology, there are blockchains today that are optimized for file storage; the platform Sia, for example, is a blockchain made to connect users who need file storage space with hosts worldwide offering underutilized hard drive capacity. These hosts can be literally anyone, even a normal personal computer, looking to rent out their unused disk space. According to the Sia website: “The Sia software will divide files into 30 segments before uploading them, each segment targeted for distribution to hosts across the world. This distribution assures that no one host represents a single point of failure and will reinforce network uptime and redundancy.” The segments are also encrypted with the Twofish algorithm, so that the only one who can read the files is the person holding the private key. This ensures that no one host can read any individual segmented piece of the file (Vorick & Champine, 2014). The only thing actually stored on the Sia blockchain is the smart contract that binds the user in need of disk space and the user renting it, thus making the blockchain quite small (SiaStats, u.d.). Other examples of blockchain file storage is Storj (Wilkinson, o.a., 2016) and Filecoin (Protocol Labs, 2018), both of which uses a similar method that of Sia.

If the digital forensic blockchain uses a similar technology like the Sia blockchain, it could be the answer to solving both network congestion issues as well as increasingly growing blockchain sizes. However, renting disk space, whether it’s from normal citizens’ computers or from cloud storage providers, laws like GDPR could make it a difficult task to handle.

### 4.1.2. Image File Hashes

When examining a digital evidence image file, the forensic toolkit used will often give a reported hash of the image in various hashing algorithms like MD5 or SHA-1 (Figure 10). This hash could potentially be uploaded to the blockchain and stored like any other piece of data.

Storing only the hash of the digital evidence image files simplifies the block size dilemma dramatically. A normal file hash doesn’t take up much space, and so a traditional block size could be utilized in this case and long-term storage issues wouldn’t normally occur. The block time wouldn’t need to be very long either, since it’s a relatively small amount of data being sent to the blockchain.
Some forensic toolkits like AccessData’s FTK will display the hash of the digital evidence image (Austin, 2013).

The usefulness of storing the image hashes in a blockchain is that the provenance of any event or action can be traced back to where it originally entered the process in question, or when you want a relatively lightweight blockchain for collecting, preserving, and validating evidence which can be strengthened with the help of the blockchain. This is especially useful in models such as chain of custody. In the chapter 4.2. Practical Implementation, a small practical implementation will be shown.

4.2. Practical Implementation

This blockchain prototype will process the image hashes provided to it. The blockchain does not represent the actual full-scale operation of a blockchain. Instead, it gives a proof of concept at implementation level of how a blockchain could work, so that the idea behind it may funnel further work into making a full-scale solution. This blockchain won’t be accompanied by any application that automatically uploads the image hashes, but instead, it will use three predefined examples of image hashes to show the work in motion. This example could be an example of a private blockchain, since it’s run on the host computer alone. See Appendix A for the full source code.
The way the blockchain works is as follows (Figure 11):

1. The application will start in the `Blockchain.java` class, where three different predefined hashes in the form of MD5, SHA-1, and SHA-256, are specified in the code. These three hashes represent the image hashes one might receive upon taking a digital forensic copy of a hard disk drive. One by one, these three image hashes are sent to the `Block.java` class via the `addBlock(Block)` method together with the integer `difficulty`. The difficulty integer is a simple representation of how difficult it will be to mine the block, but more on that in the next point.

2. Inside the `Block.java` class, there are several variables that are used to create and mine the block itself. First, the `mineBlock()` method will create a difficulty string, which is the calculation the computer needs to process before the block is mined and appended to the blockchain. This calculation is needed due to the proof-of-work to deter DDoS attacks to the blockchain, among other things. When the computer finds the correct result for the difficulty string, `mineBlock()` will call upon `calculateHash()` to calculate the hash of the current block. After this calculation, which is done in the `StringUtil.java` class, the result is sent back to `mineBlock()`, which will then append the block to the blockchain.

![Figure 11: A UML diagram of the blockchain project example.](image)

*Created by Tobias Svenblad*
3. After all three blocks have been mined, a control of the blockchain will be made in the class `Blockchain.java` in the method `isChainValid()`, which checks for inconsistencies (i.e. if any of the hashes mismatch). If any inconsistency is found, an alert will be displayed.

4. Lastly, the blockchain’s current state is printed in its entirety with the help of `StringUtil.java`’s `getJSON` method. Since there are only three results in the blockchain, the list is very small, but this could potentially be much longer when working with a real blockchain.

The complete output when running this application is displayed in Figure 12. As we can see, the blocks are mined, one by one, where the hash of the current block is printed out. Then, the blockchain consistency check is made, which comes out true (meaning, there are no fiddling in the blockchain when running it). Lastly, the current state of the blockchain can be seen with all the different values that build up the blocks.

```
Mining the first block... Block has been mined! The hash of the block is: 0000000000000000000000000000000000000000000000000000000000000000
Mining the second block... Block has been mined! The hash of the block is: 0000000000000000000000000000000000000000000000000000000000000000
Mining the third block... Block has been mined! The hash of the block is: 0000000000000000000000000000000000000000000000000000000000000000
Is the blockchain valid?: true
```

```
The blockchain's current state: [
  {
    'hash': '0000000000000000000000000000000000000000000000000000000000000000',
    'previousHash': '0000000000000000000000000000000000000000000000000000000000000000',
    'data': 'Image Hash (MDS): b7318050b30d90735336490c8d6ccdebe',
    'timestamp': 15263977724529,
    'nonce': 512648
  },
  {
    'hash': '0000000000000000000000000000000000000000000000000000000000000000',
    'previousHash': '0000000000000000000000000000000000000000000000000000000000000000',
    'data': 'Image Hash (MDS): 041d7c14263b82f74f7c4a1ae1a73c4d7a88da31',
    'timestamp': 15263977724813,
    'nonce': 950663
  },
  {
    'hash': '0000000000000000000000000000000000000000000000000000000000000000',
    'previousHash': '0000000000000000000000000000000000000000000000000000000000000000',
    'data': 'Image Hash (MDS): 211ccf0b9f1d2efef9e2a2d9086b2b46b764c07123e077e270ed9e8e9775',
    'timestamp': 15263977726490,
    'nonce': 424688
  }
]
```

**Figure 12**: The result output when running the practical example.

The execution time when mining the blocks depends on the available computer power. Mining the blocks with a HP EliteBook 840 G3, Intel Core i7-6500U @ 2.50 GHz is quite fast, around three seconds for each block when using a difficulty integer of 5. Increasing this number to 10 will cause the block to be mined after around 50 minutes. If we were to upload the whole evidence files, as discussed in chapter 4.1.1. Whole Evidence Files, this time would theoretically increase.

### 4.3. Suggested Application

Allowing the digital forensics to communicate with the blockchain is a very important step that shall not be overlooked. The previous chapter, 4.2. Practical
Implementation, isn’t very user friendly and would need some type of graphical user interface so it doesn’t feel too cumbersome to handle. Creating a special application to upload either the digital evidence files themselves, or only the image hashes to the blockchain, would allow investigators an ease-of-access usage to the blockchain.

Two different applications are presented below that may suit the needs for processing and storing digital evidence on blockchain solutions:

1. The application could be developed as a stand-alone interface. This application would act as a communicator with the blockchain, allowing the investigator to upload material to it. This would be a simpler solution than the second option below, and would probably suit file image hashes. The hashes would have to be uploaded manually, but it wouldn’t require too much effort from the investigator (after all, uploading an image hash wouldn’t take very long in contrast of how long one could spend examining the evidence in question). The downside of manual labor however is the human error factor that needs to be considered.

2. The second suggestion is an application which has a combined interface for blockchain communication and a forensic toolkit for evidence investigation. This application would be much more complex than the first one and would require quite a lot of work. The application could have its own blockchain pane where one could drag-and-drop different evidence files to upload the evidence files themselves. The picture titled Figure 13 could be a suitable candidate as an application layout where the numbers represent the different modules, as they’ll be called below.

The application presented below would have the following modules:

1. This module would display the case details; the case number, case name, name of investigator, etc.
2. An evidence tree pane with the logical folder structure.
3. The file list pane - all files that are contained inside the selected folder from the evidence tree pane.
4. The viewer pane - to switch between different views like HEX format or live view of images, videos, and others.
5. File properties of the file selected. HEX interpreter values could also be displayed here.
6. The blockchain communicator - upload the evidence in this pane to send the case details and the digital evidence to the blockchain.
Figure 13: An example of how the application could look like.

After the digital forensic investigator has found a piece of evidence (e.g. an image) he could drag it via a drag-and-drop system to the blockchain pane. The evidence would be sorted into a category to keep all evidences separate (e.g. images, videos, documents, etc.) based off their file endings. If the file doesn’t have a file ending, the investigator could create his own categories or it would simply default to a generic unknown file type category.

Once the evidence is uploaded, it will automatically be propagated to all other nodes in the network. The forensic investigators sitting across the nodes would have the possibility to go into this case and browse the different digital evidences without changing anything. Making sure no one makes any illicit additions to a case, you would have the nodes voting for the appending block.

The tricky part is how the application will manage to open the images and fish out the individual files or the metadata. The different file extensions or drive partitioning systems could turn out to be an extremely complicated task. Likewise, the indexing of said files must be considered. The blockchain could, for example, create a case number in which the individual evidence files are given a unique index. But this problem might be difficult to solve nonetheless.
5. Comparisons

Although it may be difficult to compare traditional forensic methodologies with blockchains, an attempt has been made to gather theoretical information from different sources as well as comparing the, albeit small, practical implementation with real life examples.

5.1. Security

When it comes to security, two different kinds will be discussed in this chapter; physical security and digital security.

The digital security, in the aspect of processing digital evidence, are difficult to justify for either blockchains or traditional digital forensic processes. Starting off with traditional methods, as already discussed previously in this paper, there are quite a few things to watch out for. After the evidence has been replicated from the original device to the investigator’s computer, to a shared storage, or to a cloud storage, the evidence could be accessed and changed by system administrators, cloud providers, crackers, and colleagues, if the security infrastructure is compromised (e.g. by a bug or by carelessness). This could threaten the evidence integrity; if the digital forensic is not aware that a change in the evidence has been made it may easily go unnoticed.

On the other hand, countermeasures are already in place by some digital forensic tools to solve this problem. In tools such as AccessData’s FTK (AccessData, u.d.) or Guidance Software’s EnCase Forensics (Guidance Software, u.d.) there are certain capabilities to detect if the evidence has been altered as long as the evidence is saved to a logical case folder on the computer. This way, when opening the case again after the alteration has taken place, the tool will send out an alert, warning the user that the hash fingerprint between the original evidence and the altered evidence doesn’t match anymore. However, some tools don’t offer such protection, and the digital forensic might not notice the warning message, or the system could be faulty to not display the message in the first place.

When it comes to the digital security of blockchains, the main issue lies within the application hosted on the blockchain itself. The fundamental property of blockchains is that they are secure by design, but that won’t do any good if the application has a lot of vulnerabilities. A properly programmed application to communicate with the blockchain will make it more secure. For instance, the blockchain Ethereum had a vulnerability in its application that rendered 3,759 contracts and $169 million USD worth of cryptocurrency inaccessible in July 2017 (Houser, 2018).

Depending on how the blockchain is made and created, it could be possible to inject bad code into the software itself. The practical example shown in chapter 4.2. Practical Implementation shows that even the simplest blockchain is susceptible to this sort of attack unless you have other nodes with the original, unchanged software.

Blockchains are also susceptible against the ‘51% Attack’; an attack that has only been touched briefly in this paper. The premise of the attack is that if someone
owns more than 50% of all the nodes in the blockchain network, that person could essentially change and add whatever they wanted to, to the blockchain, since that person will constantly throw off all other nodes that is voting for the change. This attack doesn’t concern private blockchains however, since the organization controlling the private blockchain already owns 100% of all block creation resources, but to public and consortium blockchains, this attack could lead to devastating results (Hampton, 2016). For example, in August 2016, Krypton and Shift, two blockchains based on Ethereum, suffered from the 51% attack (Campbell, 2016).

Public blockchains also face the threat of denial-of-service (DDoS) attacks as well, since they are entirely open to the world wide web. The first recorded attack made to a blockchain was the DAO joint Ethereum investment fund in 2016, but reportedly many other blockchains have been exploited with similar attacks since then (Loeb, 2016). Such attacks could be solvable by DDoS detection and cleaning centers to wash out illicit traffic (Agarwal, Dawson, & Tryfonas, 2004).

On January 9th, 2018, Sharon Goldberg, an associate professor at Boston University; Ethan Heilman, a Ph.D. candidate at Boston University; and Yuval Marcus, a researcher at the University of Pittsburgh, published a paper titled “Low-Resource Eclipse Attacks on Ethereum’s Peer-to-Peer Network”, which is an attack which obscures a node’s view of the blockchain (Marcus, Heilman, & Goldberg, 2018). As explained in Bitcoin Magazine, “In an eclipse attack, an attacker takes control of all the connections going to and from a targeted victim’s node. This way, an attacker prevents that victim from obtaining full information about other parts of the network.” (Castor, 2018). This attack is especially useful in blockchains that handle transactions, like the Bitcoin or Ethereum network, because you can potentially spend your funds twice in a so-called double-spend attack. A payee could send his coins to recipient one, perform an eclipse attack on that recipient and at the same time, sends the same coins to recipient two. The both recipients are then unaware that the same coins were used in two transactions.

Fake DNS attacks are also a worrisome action that could take place when discovering nodes. If an attacker spoofs the DNS address or uses DNS Cache Poisoning, the DNS resolves the name of a node address to an illicit third party, potentially causing integrity issues (Kovacs, 2016).

Johnny Lee, author of Beyond bitcoin: Leveraging blockchain for forensic applications, says that the blockchain may have a potential privacy issue if the blockchain is hosted as a public ledger. He says: “The industry is looking at advanced encryption technology as a way to protect the privacy of transactions in a public blockchain ledger. This would allow only authorized people to see a transaction’s value, but the transaction could still be recorded in a public ledger. This would restrict access to information that could be used for fraud purposes, for example, while letting accounting systems and risk managers rely on the information they can see.” (Lee, 2017).

Despite this, the particularly usefulness of blockchains in a forensic context is that once you have a definitive record of transactions the auditing falls naturally easy. By extension, all types of blockchains permits ongoing auditability - auditors, as
well as fraud and forensic investigators, could perform reliable spot-checks at any time because the record would be verifiably completed as of that point in time. Blockchains also offer great security with its cryptographic protocols. It is extremely difficult to intercept the network traffic sent to the blockchain, or change anything on the blockchain without the majority of the node’s consensus. This makes a great standing point to achieve the integrity protecting platform to strive for.

The physical security of the traditional evidence process depends heavily on which method the digital forensic applies. In some places, after the investigation has been made, the copy of the original device is written to a separate device, which is then locked inside a safe or an evidence room. This practice might be safe unless the area is prone to tornadoes or floods, which could destroy the evidence. As the blockchain is digital however, there is no need to save a physical copy of the evidence anywhere; it is distributed across the nodes automatically in consortium and public blockchains. With private blockchains, there is always a possibility that if mother nature destroys the central building together with all IT equipment, the blockchain could be destroyed with it.

5.2. Performance
Performance, in terms of speed, cannot easily be measured when it comes to processing and storing digital evidence (since it all depends on the size of the evidence, the speed of the computer used, the way the evidence is treated, and different human factors). An estimate given by a digital forensic in Mariehamn, Finland, says he puts down roughly 90 - 120 hours to process and index a hard disk drive for digital evidence with a moderately fast computer (Kull, 2018).

Blockchains are probably somewhat slower than their traditional counterparts when it comes to processing digital evidence. The performance of a blockchain depends on what block size is sent do the blockchain (i.e. the amount of data sent to every other node) as well as what block time the blockchain utilized. The block time is the average time it takes for the network to generate a block; this can be anywhere from every five seconds to every ten minutes. The block time is usually connected to something called proof-of-work, which is a technique put in place so that there is always a short delay of adding or changing blocks, so that no one can change the blocks with rapid succession without anyone noticing.

This block time and proof-of-work was demonstrated in chapter 4.2. Practical Implementation, where the difficulty variable could be tuned to match the desired block time. If that was a full scaled production blockchain, there could be issues with setting the difficulty too high and too low.

In fact, banks struggle to make blockchain based applications fast enough, and it seems difficult to find the appropriate block time for various applications (Stafford, 2016). The block time, used in digital forensic purposes, has no reason to be longer than necessary, which would be helpful if speed is of utmost importance (like in a consortium blockchain where a murderer is on the loose and you need to spread the information quickly). Or, the block time could be longer if
integrity is important, and there are other parties watching the blockchain, like that of public blockchains (to mitigate a 51% attack, for instance).

Blockchains are also overall computationally expensive, and if combined with encryption, even more so. Blockchains cannot utilize multi-threading and parallelization the same way as other applications can. The reason has been explained by Andy Vitus of TechCrunch like this:

“This goes back to the distributed nature of blockchain’s architecture and the consensus mechanisms that verify activity on the blockchain. In this environment, the infinite parallel execution that comes from every node on the network computing every transaction means that compute costs are extremely high. In other words, there is very little excess compute power available to the network, making it an exceptionally scarce (and therefore expensive) resource.” (Vitus, 2018).

Vitus goes on to explain that adding more computers to the blockchain won’t solve the problem. In fact, it will do the complete opposite. The more computers (or nodes) that join the blockchain, the more nodes required to sync with the latest transaction history.

Nikolai Hampton of Computerworld said in an article that “many in-house blockchain solutions will be nothing more than cumbersome databases,” hinting that the blockchain technology might not be the best solution for all user cases and will require a lot of unnecessary man-power, time, and effort to put together (Hampton, 2016).

The conclusion is that while a traditional system might be faster, as you don’t have to wait for any evidence to be uploaded to the blockchain, the benefits might outweigh the negatives depending on the specific user case. For instance, if evidence integrity is an important must (which is often always important), then blockchains are the better alternative, while if speed is more important, the traditional methods might still be a good choice.

5.3. Cost
There are a lot of contradicting studies on the cost of blockchains. David Treat, Managing Director of Accenture Financial Services and Blockchain Lead, explains that blockchains used in the financial sector could benefit with a cost reduction of up to two-thirds with blockchains replacing legacy back-office infrastructure. “Blockchain could lead banks to decommission much of that infrastructure and externalize key operational processes. It could completely change the cost dynamics in these organizations.” (Treat & Brodersen, 2017).

Max Di Gregorio, Technology Consulting Financial Services Lead for Middle East & North Africa at PWC, a direct competitor of Accenture, stated a similar conclusion, “According to a Santander FinTech study, distributed ledger technology could reduce financial services infrastructure cost between US$15 billion and $20 billion per annum by 2022.” (Di Gregorio, 2016).
On the other hand, the money saved from deprecating certain infrastructure services might instead be spent on electricity, storage, and education for trained specialists (Smiley, 2017). Chris Haley, Corporate Strategy Director at ICF, and Michael Whitaker, Vice President of Emerging Solutions at ICF, says that “Energy costs may rise tremendously as the transaction volume increases.” (Haley & Whitaker, 2017). It is possible to see this energy cost in action in the Bitcoin blockchain. According to Blockchain Luxembourg, a single transaction of Bitcoin costs between $75 USD and $160 USD (Blockchain Luxembourg, 2018).

Moreover, the storage requirements for blockchains are ever increasing. Since there is no data retention in a blockchain, and data is immutable, it will persist through time. The individual nodes must also maintain an immutable ledger of all transactions back to the origin of each blockchain. The Interplanetary Database (IPDB) Foundation studied the storage needs for blockchains used by IoT, music streaming services and energy markets, and concluded that on premise storage would be more expensive than cloud storage, but even then, Jamila Omaar says that “A blockchain database must store data indefinitely, so the [cloud’s] recurring payment model doesn’t work,” depending on what type of blockchain application its hosting and how many nodes that are participating in the network (Omaar, 2017). For instance, having 20 nodes with 20 gigabytes worth of data on the blockchain, the resulting storage need would be around 400 gigabytes. This could work if the data was static, but as more and more transactions enter the blockchain, each transaction must be duplicated to all other nodes, making a cloud solution scalability very costly.

5.4. Scalability
Bernard Golden, Vice President of Cloud Strategy at Capital One, says that there is an issue with blockchains that makes scalability difficult. “If one node has performance, scale, or security problems, they can impact the other nodes,” he says (Allen, 2017). Chris Haley, Corporate Strategy Director at ICF, gives a little more insight into what is causing the scalability cost issue. He says that “[the] Blockchain’s energy costs and storage requirements, as well as the uncertainty that accompanies any nascent technology, make it a far less scalable solution than existing databases.” (Haley & Whitaker, 2017).

The challenges to adoption a blockchain of any type is that the business leaders struggle with the applicability of blockchain to everyday business problems. Digital forensics may see the need to solve various integrity, value and/or ownership issues of specific assets while the same issues may not have any practical applications to business leaders (Lee, 2017).

6. Discussion
The findings in this paper suggest that blockchain technology might hold the answer to securing the integrity of digital evidence. The study confirms, however, that not all blockchain models and infrastructures are mature enough to be deployed in a digital forensic live production environment for processing and storing digital evidence. The digital evidence integrity issue can be justified by considering the overall security designed in a blockchain and how they are not easily modified without anyone noticing, especially when several independent
nodes are working together. This would have created the perfect system, if it hadn’t been for some of the vulnerabilities discussed. Some of these vulnerabilities might seem exaggerated, but the fact remains that these attacks have happened in real life and they are always to be considered from a cyber security point of view.

The performance of blockchains is also questionable. Blockchains are computationally expensive, which requires a lot of processing power to handle efficiently (and to avoid congestions in the network). If the blockchain is combined with encryption, the cost is even higher. Unlocking parallelization and writing multi-threaded code is the norm today – but not so with blockchains, due to the distributed architecture and consensus protocol that needs to be met. Adding more computers to the blockchain does however not solve the problem, since all nodes needs to sync with the latest transaction history on the blockchain. The block size as well as the block time are two factors that are huge impacts of the performance as well. If the block is too large, it could take a very long time to process for all nodes. If the block time is too long, the nodes must wait longer for the blockchain to create new blocks.

Our finding that blockchains could provide the needed technology to verify digital evidence replicate the finding of Salgado’s *Blockchain of Evidence*, upon which the user scenarios in chapter 2.2. User Scenarios was based. Salgado’s study differs from this one however, since his study is based entirely off digital images and videos being distributed and edited, while this study focuses more on the general digital evidences one might find. The study conducted by Lone & Myr, titled *Forensic-Chain*, has also been discussed in this paper. It is relatable because of the heavy focus on chain-of-custody and the way the data structure would allow the creation of a digital ledger for recording and storing transactions shared by all participants in the network. The blockchain purposed by Lone & Myr is built on top of Ethereum, while this study doesn’t prefer one side to the other.

In my own opinion, blockchains show great potential but they might not be the best technology suited for storing and processing digital evidence in its current developed state. For example, if the issues of scalability, performance and security aren’t addressed well ahead of time, the blockchain could end up becoming a cumbersome database with difficulties making changes to.

I do not think, however, that these blockchain solutions should be developed by the individual police departments alone. The better alternative would be if the digital forensic industry, like AccessData (FTK) and Guidance Software (EnCase), could find a way to implement their own blockchain solution that they could sell. That way, the application part would be taken care of.

With that said, this study concludes that there is still a lot of work to do before pulling the trigger to deploy blockchain technology for processing and storing digital evidence. Small scale testing environments would be needed to develop the blockchain, and even then, active support and maintenance would be needed on a regular basis to mitigate the different attacks that has surfaced in the past. For instance, consortium blockchains might not have the required block size that is required, depending on what type of material you choose to upload to the blockchain. Likewise, if going with a public blockchain, how well does the
solution respond with GDPR or any other privacy related regulation or law? There is a lot of consequences one will have to consider, as it is with all relatively new technologies (at least as a mainstream thing).

Blockchains are very complex and will require specially trained personnel to build and develop, which could lead to huge costs for the company. Electricity and hardware also won’t come cheaply, which is probably why most blockchains today operate on public nodes, which ultimately acts as free hardware to mine the blocks (as is the case with Bitcoin or Ethereum). If the digital forensic blockchain was to be of a private or consortium model, that processing power needs to come internally, which could lead to quite a massive cost increase. Of course, building a blockchain would make it possible to replace or deprecate old hardware, saving some costs in return.

7. Conclusion
In summary, this study has aimed to provide useful information and technological details as to measure the usefulness of blockchain technology when processing and storing digital evidence, as well as weight the benefits against the drawbacks of implementing such a solution.

The way blockchain solutions could be used to process and store digital evidence is manifold and only time will tell if it is a suitable technology for said purpose. The drawbacks of blockchains are weight more than the benefits as of time of writing, and without well-qualified existing studies of full-scale blockchain implementation, there’s no telling if it would work well in practice.

At the time of writing, no practical research involving blockchains and digital evidence can be found online.

8. Further Research
There are plenty of practical applications for the results of this study. Further work includes building a practical, larger blockchain to use for processing and storing whole evidence files. The performance, security and scalability of such operation would require a lot of work and resources; network, storage, programming, processing power, to name a few.

The blockchain communication application would also be an interesting project to follow. How would that look like practically? Would it be possible to build one that is integrated with the overall layout HUD of the forensic toolkit? A possibility would be to create it as a plugin to EnCase or any other forensic application. Another interesting point to research is how the evidence indexing would look like once uploaded to the blockchain.

Finding out which blockchain model works best for this type of scenario is also something which would need to be improved upon. The theoretical assumptions don’t necessarily have to correspond to its practical use case, and thus, may render some blockchain types unsuitable for processing and storing digital evidences. A
more detailed measurement of speed, cost, security, and scalability of said models would also be something to further research practically.

GDPR, NIS and other regulations might stop or hinder some information from spreading node-to-node, and is worth considering, especially for chain of custody models. If GDPR poses a threat for spreading case information, what would be a suitable replacement for said information?
References

37


Leyden, J., Claburn, T., & Williams, C. (2017, February 23). *'First ever' SHA-1 hash collision calculated. All it took were five clever brains... and 6,610 years of processor time*. Retrieved April 18, 2018, from The Register: https://www.theregister.co.uk/2017/02/23/google_first_sha1_collision/


Appendixes

Appendix A
The following source code is a modified version of a blockchain creation tutorial created by Kass from Medium, which can be found on GitHub here: https://github.com/CryptoKass/NoobChain-Tutorial-Part-1

Blockchain.java
package digitalEvidenceBlockchain;
import java.util.ArrayList;

public class Blockchain {

    public static ArrayList<Block> blockchain = new ArrayList<Block>();
    public static int difficulty = 5; // A integer that determines the difficulty to mine a block.

    public static void main(String[] args) {

        // Add our blocks to the blockchain ArrayList:
        System.out.println("Mining the first block...");
        addBlock(new Block("Image Hash (MD5): b731885853bb5057338d48cc8bee", "0"));

        System.out.println("Mining the second block...");
        addBlock(new Block("Image Hash (SHA-1): 913027c14263a825748541ae1a713eb7a860a331", blockchain.get(blockchain.size() - 1).hash));

        System.out.println("Mining the third block...");
        addBlock(new Block("Image Hash (SHA-256): 31ccf6b97bcebe5f86a13b57686294d87d4cd7513ad677a577ad9beb8a3973", blockchain.get(blockchain.size() - 1).hash));

        System.out.println("\nIs the blockchain valid? : " + isChainValid());

        String blockchainJson = StringUtil.getJson(blockchain);
        System.out.println("\nThe blockchain's current state: ");
        System.out.println(blockchainJson);
    }

    public static Boolean isChainValid() {
        Block currentBlock;
        Block previousBlock;
        String hashTarget = new String(new char[difficulty]).replace('0', '0');

        // Loop through the blockchain to check the block hashes:
        for (int i = 1; i < blockchain.size(); i++) {
            currentBlock = blockchain.get(i);
            previousBlock = blockchain.get(i - 1);

            if (!previousBlock.hash.equals(currentBlock.previousHash)) return false;

            // Calculate the target hash and check if the current block's hash matches
            if (!currentBlock.hash.equals(hashTarget)) return false;
        }

        return true;
    }

    public static void addBlock(Block blockchain) {
        blockchain.previousHash = StringUtil.getHash(blockchain); // Initialize previous hash
        blockchain.hash = StringUtil.getHash(blockchain, difficulty); // Set the hash
        blockchain.timestamp = new Date().getTime(); // Set the timestamp
        blockchain.chainIndex = blockchain.blockchain.size(); // Set the index

        // Add the block to the blockchain
        blockchain.blockchain.add(blockchain);
    }
}

String StringUtil.getHash(String string) {
    // Hashing logic
}
currentBlock = blockchain.get(i);
previousBlock = blockchain.get(i - 1);

// Compare registered hash and calculated hash:
if (!currentBlock.hash.equals(currentBlock.calculateHash())) {
    System.out.println("Current Hashes not equal");
    return false;
}

// Compare previous hash and registered previous hash:
if (!previousBlock.hash.equals(currentBlock.previousHash)) {
    System.out.println("Previous Hashes not equal");
    return false;
}

// Check if hash is solved:
if (!currentBlock.hash.substring(0, difficulty).equals(hashTarget)) {
    System.out.println("This block hasn't been mined");
    return false;
}

return true;

}

public static void addBlock(Block newBlock) {
    newBlock.mineBlock(difficulty);
    blockchain.add(newBlock);
}

}

Block.java
package digitalEvidenceBlockchain;

import java.util.Date;

public class Block {

    public String hash; // The hash of the current block.
    public String previousHash; // The hash of the previous block.
    private String data; // Our data in the form of digital evidence image hashes.
    private long timeStamp; // A timestamp in the form of milliseconds since 1/1/1970.
    private int nonce; // The nonce, an arbitrary number that can only be used once.

    // The Block Constructor.
    public Block(String data, String previousHash) {
        this.data = data;
        this.previousHash = previousHash;
        this.timeStamp = new Date().getTime(); // Fetches the current time.
this.hash = calculateHash(); // Making sure we do this after we set the other values.
}

// Calculate new hash based on blocks contents which is sent to the StringUtil class.
public String calculateHash() {
    String calculatedhash = StringUtil.applySha256(
        previousHash +
        Long.toString(timeStamp) +
        Integer.toString(nonce) +
        data
    );
    return calculatedhash;
}

// Increases nonce value until hash target is reached.
public void mineBlock(int difficulty) {
    String target = StringUtil.getDifficultyString(difficulty); // Create a string with difficulty * "0"
    while (!hash.substring(0, difficulty).equals(target)) {
        nonce++;
        hash = calculateHash();
    }
    System.out.println("Block has been mined! The hash of the block is: " + hash);
}

StringUtil.java
package digitalEvidenceBlockchain;
import java.security.MessageDigest;
import com.google.gson.GsonBuilder;

import com.google.gson.GsonBuilder;

public class StringUtil {

    // Applies SHA-256 to a string (the block) and returns the result.
    public static String applySha256(String input) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");

            // Applies SHA-256 to our input.
            byte[] hash = digest.digest(input.getBytes("UTF-8"));

            StringBuffer hexString = new StringBuffer(); // This will contain the hash as Hexadecimal

for (int i = 0; i < hash.length; i++) {
    String hex = Integer.toHexString(0xff & hash[i]);
    if (hex.length() == 1) hexString.append('0');
    hexString.append(hex);
}
return hexString.toString();

} catch (Exception e) {
    throw new RuntimeException(e);
}

// Short hand helper to turn object into a JSON string for easier formatting.
public static String getJson(Object o) {
    return new GsonBuilder().setPrettyPrinting().create().toJson(o);
}

// Returns difficulty string target to compare to hash. A difficulty of 5 will return "00000"
public static String getDifficultyString(int difficulty) {
    return new String(new char[difficulty]).replace('0', '0');
}