OULU UNIVERSITY OF
APPLIED SCIENCES

Tijmen Bult

Security analysis of blockchain technology

Analyzing security issues in the context of blockchain, cryptocurrencies and distributed storage solutions

Security analysis of blockchain technology

Analyzing security issues in the context of blockchain, cryptocurrencies and distributed storage solutions

Tijmen Bult
Security analysis of blockchain technology
Spring 2019
Business Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Business Information Technology

Author(s): Tijmen Bult
Title of Bachelor´s thesis: Security analysis of blockchain technology
Supervisor(s): Teppo Räisänen
Term and year of completion: Spring 2019 Number of pages: 41

This thesis was commissioned by a financial management company. This company handles big amounts of sensitive data of clients which eventually needs to be transferred to them. The company recognizes that the technology is nascent and is therefore concerned about the security of it.

The objective of this thesis is to lay out the rough basics of the technology, explore the known security risks that exist, and determine if it is feasible to use blockchain for facilitating permanent data storage while offering a convenient way of transferring data.

As the technology is fairly new, there is not enough scientific literature available to answer these questions by using mainly scientific sources such as books or research reports. By using internet sources, the latest information can be used to formulate an idea of the current state of the sector, technology, and feasibility.

While it quickly becomes clear that blockchain in its purest form is not an ideal basis to store large amounts of data, there is an alternative. Protocols like Storj might be fitting solutions to the needs of this company.

While there are no big enterprises known to use this exact protocol to store sensitive data, the fundamentals of the protocol make it seem like there is a possibility is could fulfil the wishes of the company. However, extensive testing is needed to draw a final conclusion.

Keywords: Blockchain, data, security, archiving, storage, distributed ledger technology, data transfer

CONTENTS

1 INTRODUCTION

We are on the brink of the so called “fourth industrial revolution”. Building on the third industrial revolution, which used electronics and information technology to automate many manufacturing processes, the fourth will fuse many technologies together. Artificial intelligence, IoT, and blockchain technology are among the technologies which can completely change the world as we know it.

Blockchain is a new technology that offers the possibility to create a distributed database that is maintained by thousands of nodes. These nodes all make sure that the data in the blockchain is safe and that the data will be stored forever.

Many blockchain projects arose during the ‘cryptocurrency boom’ in the end of 2017 and the beginning of 2018. We saw many promises, of which many never came true. Blockchain therefore is often looked at with certain skepticism. But as some of the promises are coming true, the outlook of the technology seems to become more positive every day.

This thesis is commissioned by a financial management company that is interested in using blockchain technology for multiple use cases within its processes. In consultation with this company the author decided to first cover blockchain technology in its basic form, then explore the security risks of the technology, and finally to dive deeper into storage solutions that are based on blockchain technology or certain aspects of it.

A big part of the activities of a financial management company is storing sensitive financial data for its customers. In most cases these files must be stored for long periods of time, and after this period has ended the files need to be transferred to the customer in a safe but convenient manner.

By using mostly internet sources, the latest information about this sector can be used. This way, an up to date overview can be created to get a better idea of what the technology can do and how it can innovate processes such as data archiving and transferring.

In the first chapter the absolute basics will be explained, after which security issues will be investigated. In the final chapter a brief introduction of decentralized data archival will be given with the goal of determining the feasibility of using this technology in real world use-cases of an enterprise.

2 BLOCKCHAIN

What is blockchain, how does it work, and how is it already implemented? In this chapter the technology and its components will be introduced. After the short introduction, a couple of projects that implement blockchain will be explored. This chapter should lay the basis on which further exploration can be done.

2.1 Basic implementation

Blockchain technology today has many different implementations. Therefore, as Mougayar (2016, 20) states, it is not a standalone product that can be switched on or off. It forms the basis of other products, much like the internet forms the basis for the world wide web. To be able to understand the risks that exist around the use of blockchain technology, the basics of the technology need to be understood.

Mougayar & Buterin (2016) describe that blockchain is "at its core a technology that permanently records transactions in a way that cannot be later erased but can only be sequentially updated, in essence keeping a never-ending historical trail.

One of the most common implementations of blockchain technology therefore is as a so-called *tamper-evident log*. This is a structure in which data can be securely stored in a chain of data blocks. Every part, or block, of the chain is linked to the previous block. This means that when an entity tries to change data in an earlier block, it will break the chain and thus it will be detected (Narayan et al., 2016)

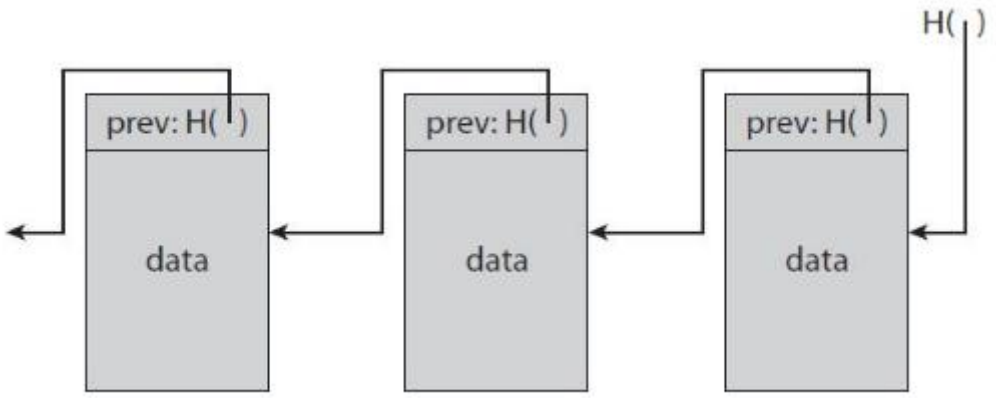

Figure 2-1 Blockchain as a tamper-evident log (Narayan et al. 2016).

As seen in figure 2-1, every block is connected with each other with the help of a *hash pointer*. This is a cryptographic hash that points to the previous block. It is important to understand that these hash pointers are the output of the data of a block header to which a hash function is applied.
Since every block in the chain is connected by these hash pointers, the smallest changes of data in a previous block can break the link, as it changes the hash pointer. Therefore, when data is tampered with, this will be detected immediately. The important take away of this first paragraph is that a blockchain consists of blocks of data, which are linked to each other by a cryptographic hash pointer that if changed, breaks the chain of blocks.

2.1.1 Cryptographic hash functions

For blocks of data to be linked with each other in a chain, a hash needs to be generated. Using a cryptographic hash function, any size of data can be transformed into a string of seemingly random numbers and letters. The size of the input data can be of any size, but the output is always a string with a fixed size (Narayan et al., 2016).
There are many algorithms that can produce different types of hash codes in their own ways. To get a better understanding of how these hash functions help secure data and link blocks in a blockchain, three examples will be given which show how a small change of the input data can produce a completely different output. In the example the *SHA-256* algorithm will be used.

Table 1 Example of hashing algorithm hashing a string of text

Input	**-->**	**Output**
Hello World	SHA256	A591A6D40BF420404A011733CFB7B190D62C65BF0BCDA32B57B277D9AD9F146E
Hello World!	SHA256	7F83B1657FF1FC53B92DC18148A1D65DFC2D4B1FA3D677284ADDD200126D9069
Hello World.	SHA256	F4BB1975BF1F81F76CE824F7536C1E101A8060A632A52289D530A6F600D52C92

In table 1, the left column represents the input and the right column the output. When this input is put through the SHA256 hash algorithm an output of 256 bits is produced. All the outputs are exactly

64 characters long, no matter the character count of the input. By adding a dot or an exclamation mark, the output that is produced by the SHA256 algorithm changes completely. The fact that this changes altogether is an important part of the miner's work on a blockchain network.

2.1.2 Mining and nodes

When looking at a blockchain network that implements the *proof-of-work protocol (PoW)*, a miner must put in effort and energy to mine a block and validate transactions. PoW protects these networks, such as the Bitcoin network, from hackers attacking the network wanting to change data in the blockchain. To validate a block, it costs a lot of energy and therefore money to solve the mathematical problem, which needs to be solved in order to validate a block. If an attacker would want to manipulate data on the Bitcoin blockchain for example, he or she therefore needs a tremendous amount of computing power to make an attack successful, making it not worth doing it as the costs of such an attack far outweigh the profits he or she potentially could gain (Tarr 2018., cited 28.02.2019).

As mentioned in the first paragraph; hash pointers are the output of the data of a block header to which a hash function is applied. In other words, the information that is in the header of each block is put through a hash function. The information that is stored in a block header is in Bitcoin's case:

Table 2 Information in block header in the case of Bitcoin (Frankenfield 2018, cited 11.03.2019).

The version number of the Bitcoin software	The current version of the Bitcoin protocol that is being implemented by the participating network nodes.
Previous block hash	The hash of the previous blockchain header. This links the current and previous block together.
Merkle root	A data structure that uses the hash codes of transaction data to create a tree of hash codes that are all connected with each other. The Merkle root is the hash code to which all these other hash codes are connected (see figure 2-2 for a visualization of a Merkle tree).

Difficulty target	The difficulty target set for the miners.
Nonce	A number that a miner can change in order to find the difficulty target hash.

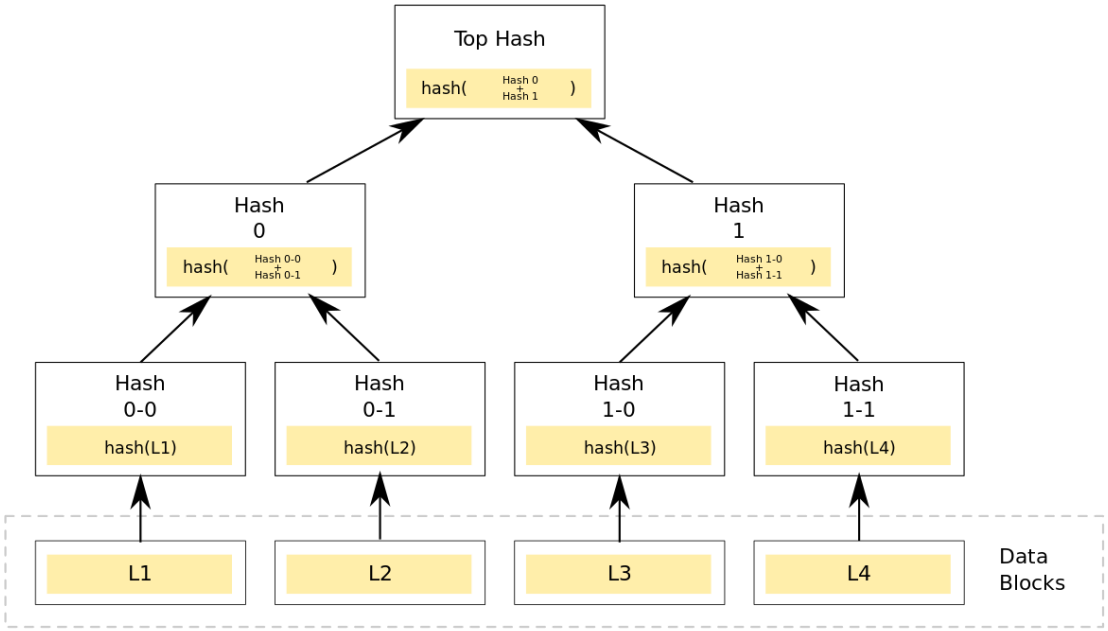

Figure 2-2 A Merkle tree data structure. L1, L2, L3 and L4 represent individual transactions (Azaghal 2012, cited 07.05.2019).

It is the miner's job to verify transactions and add them to a block. As blockchains have a maximum block size, the miners fill a block up to the maximum size with transaction data and start applying a hashing algorithm. The miners spit out numerous hashes to try to guess the right hash. The right hash in the context of the Bitcoin blockchain is a hash that starts with the right amount of zeros. This right amount of zeros is decided on the basis the amount of hashing power there is on the bitcoin network; the more miners are participating on the network, the more zeros the target hash needs to start with, the more difficult it is for a miner to guess the right hash (Good Audience 2018, cited 11.03.2019).

To guess the right hash, a miner must therefore try many different hash inputs. It does this by changing the block header data. As seen in the paragraph on cryptographic hash functions, an entire hash code can change by only changing one character. This is where the *nonce* comes in, a random number that a miner can change in order to change the hash outcome (Frankenfield 2017, cited 11.03.2019).

Simply put, a miner tries many different nonces (random numbers) in combination with the other data that is part of a block header to generate hash codes. As soon as a miner guesses the right hash code that meets the set target hash, the block is sent to other miners. They verify that the miner that says it found the right nonce did indeed find the correct nonce. When a majority of the miners says it is correct, the block is distributed among all the nodes that are part of the blockchain network, which means that the block is added to the blockchain (Good Audience 2018, cited 11.03.2019).

All the blocks that are verified by miners are distributed among all the nodes on a blockchain network. Miners themselves are masternodes, which means they store the complete version of the blockchain history. Miners need to be masternodes in order to verify transactions by comparing the transactions with the total transaction history of a blockchain, so that they can determine if an entity has sufficient funds to send the transaction. A masternode doesn't have to be a miner. A masternode can also be a computer on which the complete data set is just stored (Jimi, S. 2018, cited 11.03.2019).

Nodes are the foundation of blockchain technology. All the data stored on a blockchain is distributed among a network of nodes which realizes the decentralization of a network. In case a node is hacked or has a technical malfunction, the data set is still stored on the other nodes. Therefore, it is more resilient than a centralized database (World Crypto Index 2019, cited 11.03.2019).

2.2 Implementations of blockchain technology

As stated in the first paragraph, blockchain is not a standalone product that can be switched on or off. It forms the basis for other solutions, much like the internet forms the basis for the world wide web. In this paragraph two implementation of blockchain technology are explored; Bitcoin and Ethereum. Also, the difference between public and private blockchains will be shortly explained.

2.2.1 Bitcoin and its distributed ledger

Bitcoin was the first peer-to-peer platform to utilize blockchain technology to facilitate payments. Important for Nakamoto, the inventor and developer of the protocol, was that the network would not be dependent on a third party to avoid double spending. Or as Nakamoto (2008) writes in Bitcoin's whitepaper: "A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution." He goes

on to say: “We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.”

In 2009 the “Genesis Block” was formed, the first ever block of the Bitcoin blockchain. Since then, the value of bitcoin, the cryptocurrency of the Bitcoin network, has risen to a record height of almost $20.000 (Futurism 2019, cited 30.04.2019). Due to the rapid growth in value, bitcoin has been an interesting investment opportunity for many investors to speculate with but is often also called the “digital gold” or “the money of the internet.”

Bitcoin uses a distributed ledger in which nodes keep track of all the transactions that have happened. These transactions are added to blocks by miners and are transformed into hash codes. This way, not only all the blocks in the blockchain are connected, also the transactions themselves are connected to each other. In practice this means that when only one small detail is changed in a transaction, the whole block will be invalid. Bitcoin manages to create a trustless network this way, in which no third party is needed to check if transactions are valid (Nakamoto 2008).

Bitcoin is slowly being adapted by big traditional companies such as Microsoft and travel agency Expedia (Barton 2019, cited 30.04.2019). The network is, however, running into scalability problems. This is mainly due to the fact that the creation of blocks with around 2.000 transactions takes on average 10 minutes, and also because there is a block size limit. In practice this means that only a limited amount of transactions can be processed, leading to slower processing times when the network activity is high. The block size limit is, however, a defense mechanism against DOS attacks to prevent hackers from making very big blocks which would disrupt the Bitcoin network (Aziz 2019, cited 30.04.2019).

Other blockchain projects that use blockchain as a distributed ledger to transfer value are Ripple (XRP), Bitcoin Cash (BCH) and Litecoin (LTC). Bitcoin remains the biggest cryptocurrency based on market capitalization to this day, with a market capitalization of $94 billion (CoinMarketCap 2019, cited 30.04.2019).

2.2.2 Ethereum and its smart contracts

Where Bitcoin provides a peer-to-peer blockchain network solely to send money from one person to another, Ethereum goes a step further. Ethereum was launched in 2015 and according to one of

the founders, Vitalik Buterin, the platform is supposed to be a solution to Bitcoin's limited functionality. Buterin sees Bitcoin as a "pocket calculator that can do one thing, but it does it well" (Zmudzinski 2019, cited 04.03.2019).

The main difference between Bitcoin and Ethereum is that Ethereum can be used to run programming code. These pieces of code are called "smart contracts", which are stored on the Ethereum blockchain. By allowing pieces of code to be stored and executed on the blockchain, developers can develop decentralized applications (dApps), which are run by the miners that are part of the blockchain network of Ethereum. They get rewarded with ETH, also according to the PoW protocol (Blockgeeks 2017, cited 04.03.2019).

On the Ethereum network three types of dApps can be executed. The first type is a dApp that basically is a simple contract that when certain conditions are met, distributes value between parties according to the contract stored on the Ethereum blockchain. The second type is a dApp in which money is also involved but needs information from outside of the blockchain. And lastly, dApps can be used for other purposes as well such as data storage (Ray 2019, cited 04.03.2019).

An example of a simple contract that is stored on the Ethereum blockchain is a property rental contract. Normally we have a paper contract that states all the conditions and terms of the rental agreement and it is signed by both the landlord and the tenant. One risk with these contracts is that the tenant must make sure they pay the rent in time. By converting the rental contract into an Ethereum smart contract most of the contract is automated.

A landlord can in this context create a smart contract and set the rules and conditions. A tenant can then sign this contract after which the contract cannot be removed or altered as it is stored on the immutable blockchain. If the conditions, such as "Tenant must pay rent every first day of the month until 2020", are met, the smart contract automatically takes the set rent amount from tenant's account and transfers it to the landlord (Naqvi 2017, cited 05.03.2019).

The second type of dApps are smart contracts that use information that comes from outside of the blockchain. Oxam in Sri Lanka, part of Oxam Novib, uses these types of dApps for so called 'microinsurances'. These are insurances that offer coverage for lower income groups. Oxfam Sri Lanka uses the Ethereum blockchain for insurances for rice farmers in the country. In this case the dApp contains conditions much like a traditional contract which are dependent on information that comes from outside of the blockchain. In this case the conditions are based on weather information that is being provided by a weather agency. If the weather is bad and leads to farmers not being able to harvest enough, they automatically get paid an amount to cover the damages (Allison 2018, cited 05.03.2019).

And lastly there is the type of dApp that doesn't per se revolve around moving money. An example of such dApp is Golem. This dApp allows you to use the computational power of nodes that are connected to the Golem network. These are mostly computers of private individuals but can also be big data centers (Golem 2019, cited 06.03.2019). Using Golem, a user can use the network for CGI rendering but also for machine learning. Therefore, Golem can be seen as a marketplace for computational power that is built on top of the Ethereum blockchain.

2.2.3 Public vs. Private blockchains

The two blockchain platforms that were mentioned as examples of blockchain implementations are both accessible and readable by anyone who wishes to do so. This means they are public blockchains and that anyone can join these networks and participate either as a node or a miner. There are, however, blockchains that offer more privacy; private blockchains.

Private blockchains are also called *permissioned blockchains*. Only parties that are invited to join can access such private blockchain (Jayachandran 2017, cited 07.03.2019). This means the control is more centralized, as where public blockchains are owned by no one and control is decentralized among the participating nodes.

Transactions on a public blockchain need to be verified by a majority of the nodes that keep track of the blockchain. The bigger the number of nodes, the safer the network. But a big number of nodes comes with a downside. As more and more nodes need to verify a transaction, a network gets slower and starts consuming more.

When we look at a private network, the number of nodes is controlled by an authority that decides who can participate and who can't. This means that the network can scale up faster and process more transactions as less nodes are needed to verify a transaction (Kingston 2018, cited 07.03.2019).

One might ask; what is the difference between a private blockchain and a shared database? While it is true that both private blockchains and shared databases consist of a network of entities that can both read and write, a shared database is centralized (Hohpe & Woolf 2004, cited 13.03.2019). Since private blockchains use cryptography and data structures, such as Merkle trees, it can be assured that non-valid transactions cannot be stored on a private blockchain. This means that private blockchains offer better levels of error checking and transaction validity compared to shared databases (Thomson 2016, cited 13.03.2019).

3 SECURITY ISSUES

As blockchain technology is gaining traction in a lot of different sectors, it is often hailed as an unbreakable technology. Due to the decentralization and encryption, many think that information stored in a blockchain is, and will always be safe. In this chapter different types of security issues regarding blockchain technology are examined to get a better idea of all the security threats there are for the most known blockchain platforms.

3.1 Blockchain: Miners and smart contracts as a case

Firstly, the security issues with regards to blockchain technology itself are explored. As part of blockchain technology itself, mining and smart contracts will be explored to understand different security risks in these areas of the technology.

3.1.1 Mining

When looking at blockchains that implement the Proof-of-Work algorithm, such as the blockchain of Bitcoin, it is decided by the miners which transactions or data are added to the chain. First, a miner finds the target hash, after which this solution, along with the data in the block, is broadcasted among the other miners. A majority then must find the same solution using the given information before this block can be added to the chain. However, another democratic majority is important within the blockchain concept; the length of the chain.
When a miner, or a group of miners, decide to take over the network to *double-spend* crypto assets, this is called a 51% attack. An important component of this type of attack is the length of the blockchain. How it works is when a malicious miner manages to legitimately mine a block, the miner can decide to not broadcast the solution to the other miners. By keeping it to him or herself, the miner can start his or her own private chain.

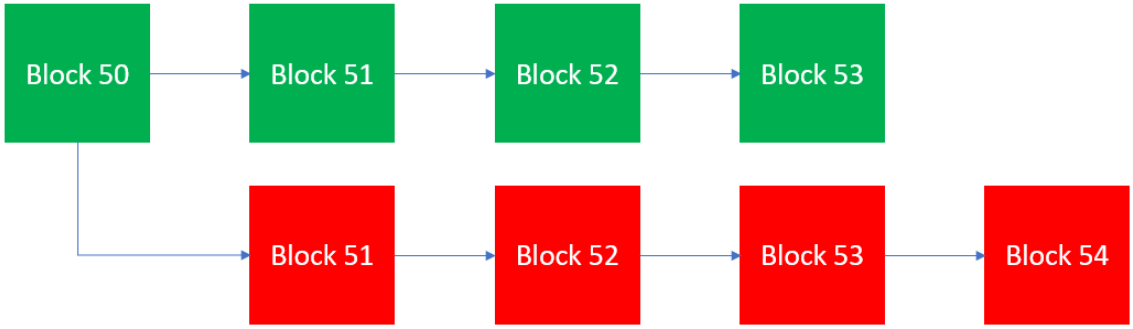

Figure 3-1 The green chain is the "legitimate" chain where the truthful miners keep adding data to, while the red chain is the chain that a malicious miner is adding transactions to, but doesn't broadcast it with the rest of the miners.

The malicious miner can keep on mining its own blocks to its own chain, while the legitimate miners still mine the original blockchain. The malicious miner then spends funds on the original blockchain but doesn't add these transactions to its own private chain. Now the miner must mine blocks faster than the original miners of the original blockchain to make the 51% attack successful.

As soon as the blockchain the malicious miner is adding blocks to becomes the longest chain (in figure 3-1 at block 54), the malicious miner broadcasts its chain to the rest of the miners, after which the blockchain protocol decides that the longer chain is the one and only chain. This happens because a blockchain that uses the Proof-of-Work algorithm follows the model of democratic governance and determines the longest chain to be the truth. Now all the transactions the miner did on what was once the only truthful chain are reversed as on its own chain it didn't do these transactions. The hacker can now double-spend these funds.

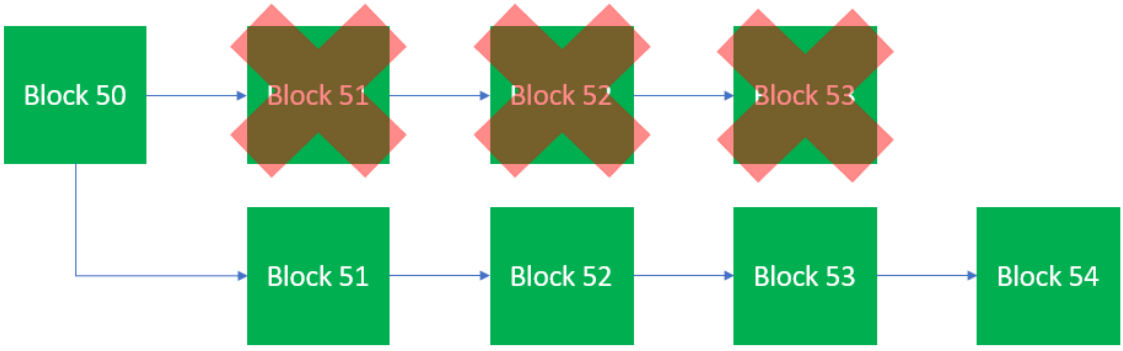

Figure 3-2 The chain that was once the only truthful chain is now illegitimate because it is not the longest chain anymore.

For a 51% attack to be successful it is essential for the malicious miner to mine faster than the rest of the miners and therefore needs 51% of the computational power of the network. It is for this reason that mainly smaller blockchain networks are targeted by this type of attacks, as the computational power of these smaller blockchains is lower due to the smaller amount of

participating miners (Jimi S. 2018, cited 13.03.2019). Ethereum Classic, a *hard-forked* version of the Ethereum blockchain, recently fell victim to such an attack. The attacker managed to double spend ETC worth $1.1 million (Orcutt 2019, cited 18.03.2019).

When a miner gains the majority of the hashing power on a blockchain network it can't, contrary to what the article on the Ethereum Classic 51% claims, change data in previous blocks, reverse transactions of other users or create cryptocurrency out of thin air. The only things a hacker can do in this case is cause a disruption within the network, prevent transactions from being confirmed, double-spend funds by reversing transactions the hacker has made him or herself (Binance 2018, cited 13.03.2019).

3.1.2 Smart contracts

As explained in paragraph 2.2.2, smart contracts are hosted on the Ethereum-blockchain and other smart contract blockchains. They are pieces of computer code that are executed when certain conditions, which are hard coded into the code, are met. Ethereum, however, is not the only blockchain that can host smart contracts. Other platforms that can host smart contracts are EOS, Stellar, Cardano and Hyperledger Fabric (Blockgeeks 2010, cited 20.03.2019).

Like software, smart contracts can contain mistakes in the code which are called bugs. In the case of normal computer software, developers can fix a bug by updating the software. But since the blockchain is immutable, as discussed in paragraph 2.1, code that is hosted on a blockchain can in no way be changed. This causes many challenges for software developers, as their software must be 100% waterproof before launching.

For example, when a developer is coding a smart contract and deploys it, this smart contract will be immutable and will always do what it's supposed to do. Any mistake will be there forever. It is therefore important to always include a 'kill switch' in the code with which a smart contract can be deactivated, this is up to the developer themselves to include this or not. How this kill switch looks like exactly is also up to the developer. (Rush 2016, cited 07.05.2019).

In the past there have been several bugs found in smart contracts. Some of them were exploited by hackers, others accidentally lead to the loss of cryptocurrency. In this paragraph the attack on gambling dApp EOSbet and the Ethereum Parity wallet bug will be investigated.

EOSbet is a dApp hosted on the EOS blockchain. It is an online casino where users can participate in classic dice throwing games. Just like any dApp, EOSbet consists of smart contracts (CCN.com 2018, cited 20.03.2019). In September 2018, the dApp was attacked by hackers who managed to

exploit a bug in the EOSbet code. Hackers were able to execute a function written in the code with a fake hash. They tricked the system this way to send EOS, the cryptocurrency of the EOS ecosystem, to wallets of the hackers. Eventually 40.000 EOS was stolen, which amounts to €130.400 at the time of writing (Canellis 2018, cited 20.03.2019).

Another bug in a smart contract that caused chaos, was not really a bug, but more a mistake. This mistake was made by the Parity cryptocurrency wallet on November 8th, 2017. The Parity cryptocurrency wallet is a cryptocurrency wallet developed by Parity Technologies and supports the ERC-20 token standard, which is the token standard of the Ethereum blockchain. Due to a rather silly mistake when deploying the *multi-sig wallets,* wallets with which you can only make transactions when multiple owners of the wallet sign the transaction, over a hundred million dollars in funds were frozen.

These multi-sig wallets consisted of two types of smart contracts: a library smart contract, and sub smart contracts. Every time a new multi-sig wallet was created by a user, a sub smart contract was initiated. These sub smart contracts were all connected with the library smart contract. The mistake Parity made here was that when they deployed the library smart contract, which contained all the crucial information with which the smaller sub smart contracts could work and can therefore be seen as the "main smart contract", they did not "initialize" it. This left this smart contract without an owner. One developer saw this and managed to initialize it, which meant this smart contract was now his. The developer decided to activate the "self-destruct" mechanism, which meant the library smart contract was deactivated. However, the sub smart contracts, meaning all the multi-sig wallets, still existed. Now that they didn't have the library smart contract anymore to communicate with, all these wallets could not be accessed anymore leading to over a hundred million in funds being frozen (Choy & Teng 2017, cited 20.03.2019).

3.2 Cryptocurrencies and the risks of using them

As blockchain technology is most commonly used as a way to transfer value, cryptocurrencies have been around for over ten years now. Users can buy cryptocurrencies in a lot of different places and they are an interesting option to invest in for some investors due to the potential high returns.

3.2.1 Cryptocurrency wallets

Another layer where vulnerabilities for a blockchain may arise, are cryptocurrency wallets. The most common implementation of blockchain technology is for a payment system, such as Bitcoin as was discussed earlier. A user who wants to use this network to send cryptocurrency needs a so-called *cryptocurrency wallet*. This is software that a user can use to monitor their balance, send or receive cryptocurrency and store private cryptocurrency keys (Blockgeeks 2017, cited 15.03.2019).

A private key is a secret number with which users can send cryptocurrency (Bitcoin.it 2019, cited 15.03.2019). In other words, it is the key with which a user can get access to their balance and sign transactions. With just this key, a hacker, for example, can access the entire wallet and then spend the currency that belongs to this private key.

A common misconception about cryptocurrency wallets is that they store cryptocurrency. This is not true, as these wallets are software that store private keys, public keys, and can on the basis of the distributed ledger determine what the balance of a wallet is (Blockgeeks 2017, cited 15.03.2019).

3.2.2 Software wallets

The first type of cryptocurrency wallet that will be explored are software wallets. A user is in this case the only one that has the private key and can decide which software wallet to use to send transactions with. This adds additional risks as there are many types of software wallets available, with some of them being malicious. The private key for a software wallet is held by only the owner of that private key (Blockgenic 2019, cited 15.03.2019).

Deemed one of the safest ways to store a private key is by writing it with a traditional pen on traditional paper. Also, a user can decide to use a dedicated “wallet generator” software. The result of using a wallet generator is a physical paper that contains the public and private key of a wallet. It is advised to completely disconnect a device when using a wallet generator to prevent hackers from intercepting the private keys (Frankenfield 2018, cited 15.03.2019).

3.2.3 Hardware wallets

Another type of wallet that is deemed to be one of the safest ways of storing private keys are *hardware wallets*. This can be compared to a regular USB-flash drive that stores the users' private keys (Bitcoin.it 2019, cited 15.03.2019). The most secure way to store private keys using these hardware wallets is by disconnecting them from the internet, which is called *cold storage* (Bajpai 2015, cited 15.03.2019). A hacker has no way to reach and attack such wallet, unless the attacker has physical access to the device. Recently, the hardware wallet manufacturer Ledger published several of such vulnerabilities it found in Trezor's, its competitor, hardware wallets. These vulnerabilities made it possible for hackers to clone a device and get access to all private keys stored on a Trezor hardware wallet (Berman 2019, cited 15.03.2019).

3.2.4 Web wallets

In the case of most cryptocurrency exchanges, a user has a so-called *web wallet* that is hosted by this cryptocurrency exchange. These exchanges have to make sure these private keys of their customers are safe (Coinbase 2019, cited 15.03.2019). There are several ways that can lead to the loss of cryptocurrency when using this type of wallet, which will be discussed in the next paragraph.

3.2.5 Cryptocurrency exchanges

Cryptocurrency exchanges are platforms where traders can buy and sell a wide variety of digital assets, or cryptocurrencies. Much like traditional stock exchanges, a cryptocurrency exchange lists many different cryptocurrencies (Cryptocurrencyfacts.com 2019, cited 18.03.2019). The biggest cryptocurrency exchange in the world today is Binance, with an adjusted trading volume of over $1 billion in the past 24 hours (CoinMarketCap 2019, cited 18.03.2019).

To be able to trade cryptocurrency on a cryptocurrency exchange, a user must first create an account. This process is very similar to creating an account for an online social media platform, for example, but in essence the user is making an account in which financial funds will be stored like a bank account and is called a web wallet, as was explained in paragraph 3.2.3. As soon as the account is created, a user can buy cryptocurrency on the exchange using a fiat-to-crypto

conversion, or send funds to the account using an own cryptocurrency wallet by sending it to the public address of the web wallet that is hosted by the exchange (Ethos 2019, cited 18.03.2019).

There are several risk factors that cryptocurrency exchanges need to deal with. One of them was discussed in the paragraph 3.2.3; the storage of private keys of the wallets of their customers. As the exchanges are responsible for the security of the private keys, a hacker with access to the storage solutions of these exchanges could potentially gain access to all, or part of the funds of the exchange's customers. On top of that, in most countries in the world cryptocurrency exchanges are barely or only partly regulated (ComplyAdvantage 2018, cited 22.03.2019). This means that relevant authorities cannot protect investors from loss of their investments as they cannot require these exchanges to comply with national regulations that do apply to other financial businesses. That cryptocurrency exchanges are in many countries unregulated is mostly caused by the fact that authorities don't agree on how cryptocurrencies should be classified. Only when a country decides on a national level if a cryptocurrency is classified as a commodity or security, it can become clear under which regulator a certain cryptocurrency falls (Kharpal 2018, cited 22.03.2019).

One example of an event where investors lost their money is the Mt. Gox hack in 2011. Mt. Gox used to be the biggest bitcoin exchange in the world. In 2011 a hacker gained access to the computer of an auditor of the platform and changed the nominal value of the bitcoin cryptocurrency on that specific exchange and also managed to transfer bitcoins that belonged to Mt. Gox customers to their own wallet. Eventually it became clear that the exchange had lost over 744.408 bitcoins, which was at the time 7% of the total bitcoin supply (McMillan 2014, cited 22.03.2019). The damage amounted to a total of €460 million and lead the exchange to go bankrupt with many customers losing their cryptocurrency funds (Norry 2018, cited 18.03.2019).

Another example of how cryptocurrency exchanges can lose cryptocurrency funds is the loss of about $190 million in user funds by the Canadian cryptocurrency exchange QuadrigaCX. The CEO of the exchange passed away in December 2018, and as he was the only one who knew the private keys, no one could access the storage devices on which these private keys were stored. An interesting detail of this case is that the CEO changed his testament 12 days before he died, even though his death came unexpected. This lead many to believe this was a so-called *exit scam*, where the CEO faked his death and ran off with the cryptocurrency (Young 2019, cited 18.03.2019). This last example shows just how important one piece of data, the private key, is.

3.3 Blockchain and the CIA triad

The CIA triad is a model with which can be determined how technology affects the confidentiality, integrity and accessibility (CIA) of data (Buckbee 2019, cited 07.05.2019). By looking at blockchain from this point of view, a better overview can be developed of how data stored on a blockchain is affected when it comes to these three data security aspects.

3.3.1 Confidentiality

According to the National Institute of Standards and Technology (2013), confidentiality means "preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information."

While it is true that all data stored in a public blockchain is accessible by anyone participating as a node, encryption of data makes sure that a potential attacker would not be able to do anything with this data. Due to use of private keys only the true owners of data can access information stored on a blockchain.

When a private key gets stolen, a hacker can access anything that is stored behind this private key. It is therefore important that private keys are backed up with devices that guarantee security.

In the future, quantum computing might cause problems for blockchain platforms that implement cryptographic private and public keys. Expected is that this technology might be able to crack the private key, something that is not possible at this moment with current computing power. This problem can be solved by developing more sophisticated hashing algorihms such as the SHA-384 algorithm. (Piscini, Dalton and Kehoe 2017).

3.3.2 Integrity

According to the National Institute of Standards and Technology (2013), integrity means "the property that data has not been altered in an unauthorized manner. Data integrity covers data in storage, during processing, and while in transit."

Blockchain technology makes it possible to create a tamper evident log, as mentioned in chapter 2. With hash codes, that completely change when even the smallest detail is altered, the integrity of data can be secured. On top of that, blockchain technology makes it possible to show the exact location where data might have been altered by an attacker.

This characteristic makes blockchain technology especially fitting for transaction data. It can be determined if data is valid or if it has been manipulated and it makes it possible to create a trustless network. Besides that, every piece of data that is added to a blockchain must be signed with a private key which makes traceability easy (Piscini, Dalton and Kehoe 2017).

3.3.3 Availability

According to the National Institute of Standards and Technology (2013) availability means "ensuring timely and reliable access to and use of information."

Data stored on a blockchain is stored by a distributed network of nodes. This means that if an attacker aims to disrupt the network by taking down one or only a small portion of the network, it will have no effect. A node under attack can simply be excluded from the network, which makes the network extremely resilient against attacks such as a DDOS attack, where an attacker tries to disrupt the network by overflowing it with a large number of transactions at once.

The decentralization of a blockchain network assures that the network has no single point of failure. This means that, for example, when Finland is hit by a nation-wide electrical outage none of the nodes in Finland can respond. But since the blockchain network, if decentralized and big enough, also has nodes in most of the other countries in the world, a full copy of the ledger will always be available.

One of the vulnerabilities of blockchain technology, however, remains the possibility of force majeure events such as a worldwide power outage (Piscini, Dalton and Kehoe 2017).

4 DECENTRALIZED DATA ARCHIVING

During this research on the topic of blockchain technology, it became clear that using blockchain in its purest form as a storage solution is not feasible. As written Chapter 2 Blockchain basics, all nodes in a blockchain network need to have the same blockchain stored on their machine in order to participate in the network. This means that all data is duplicated so that it can be distributed. When it comes to very small pieces of data that need to be stored in chronological order, such as transaction data like “(Public key A) sent 1 BTC to (Public key B)”, this doesn’t form a big problem.

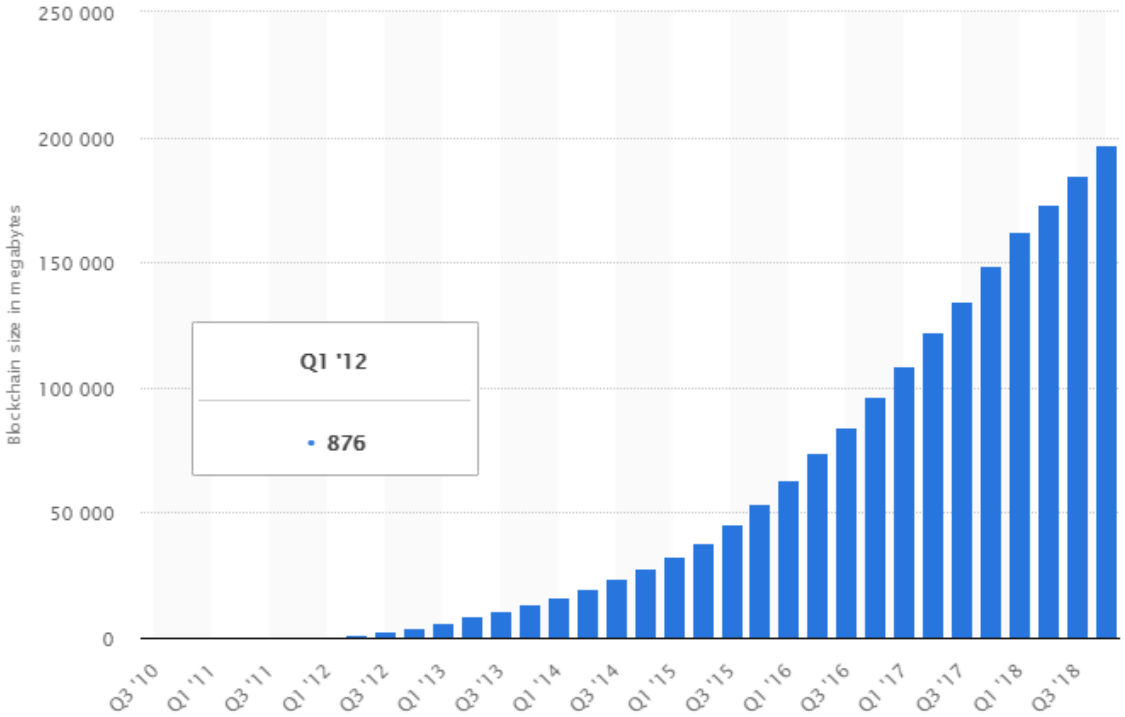

Figure 4-1 "Size of the Bitcoin blockchain from 2010 to 2019, by quarter (in megabytes)” (Statista 2019, cited 04.04.2019).

However, as shown in figure 4.1, a blockchain gets bigger over time as all previously stored data is immutable and cannot be deleted. All transactions from the past are therefore stored, which leads to a blockchain increasing in size as more and more blocks are added to the chain.

There are, however, a number of solutions that come close to the security of blockchain technology by the means of distributed storage. In this chapter such a storage solution, the Storj storage protocol, will be introduced.

4.1 The Storj protocol

First of all, what is distributed storage? BusinessDictionary (2019, cited 08.04.2019) defines distributed storage as “a computer networking scheme in which the primary objective is to pool the storage capacity of all connected devices”. Much like in a blockchain network, participating nodes are connected with each other to utilize their resources. In this paragraph the Storj protocol will be introduced with a short explanation of how the network works and how data is stored. All information will be based on the Storj v3 whitepaper (Storj Labs, Inc. 2018, cited 08.04.2019) and the conversation that the author of this thesis had with the Storj support team through several e-mails.

4.1.1 The network

The Storj network consists of three peer classes; storage nodes, satellites and uplinks. Each one of these peer classes fulfills their own role. Together they form a network of nodes that communicate with each other so that the storage of data happens in a safe but fast manner. Storj’s primary goal is to offer a “performant, secure, private, and economical cloud storage solution”. Storj encrypts, segments and distributes files across many storage nodes, eliminating the need of a central controller. As the Storj team states in an email (Leonov, A. 2019. Storj Labs Support, Storj Labs. E-mail message on 03.04.2019) “A company can build a platform with Storj network as backend. The satellite(s) will be used as backend service for such platform. You can use uplink CLI, libuplink, S3 gateway with `aws s3` CLI to manage buckets, files and permissions within your platform.”

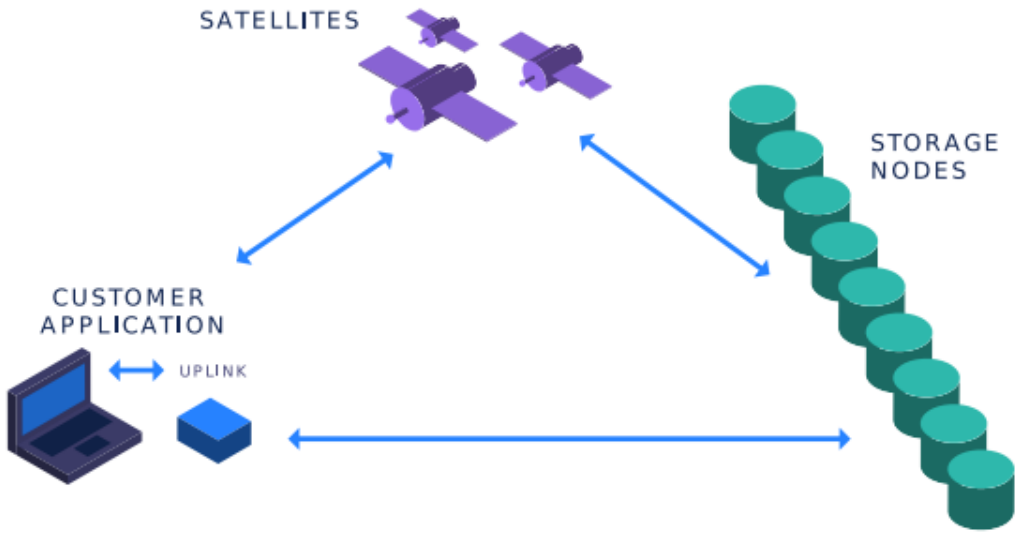

Figure 4-2, peer classes Storj protocol (Storj Labs, Inc. 2018, cited 08.04.2019)

Storage Nodes

Storage nodes can be any device with a storage capacity and an internet connection that runs the Storj protocol. Storage nodes are rewarded for the storage space they offer, their uptime and offered bandwidth. The task of a storage node is to store data and return data when requested. Satellites audit these nodes to both determine their reputation and to determine the redundancy of the data that is stored in the network. A storage node is chosen based on ping time, latency, bandwidth, storage space, geographic location and uptime among other factors. When a storage node is chosen to store data, the satellite keeps track of the address of this node by the means of encrypted metadata. A storage node can become a candidate by performing a certain task that takes effort and time. Here Storj implements a proof-of-work protocol to determine if a storage node is invested enough to be a trustworthy participant of the network. When a storage node does not perform as expected, it can be excluded by a satellite so that it won't be chosen anymore for storing data.

Satellites

Users can make an account on satellites they trust, as satellites keep track of the reputation of storage nodes they can use. Satellites also cache address information of storage nodes, store metadata with regards to file paths, and take care of the payments for these storage nodes. A satellite can be run by any user but are often elected by users. They serve as a party that takes care of the coordination of the Storj network. They basically do the back-end work for the Storj network.

Uplink

An uplink is any application that uses *libuplink,* the library of the protocol. This peer class takes care of the encryption of the data and communicates with other peer classes. The uplink can be seen as the front-end of the Storj protocol. The uplink retrieves an encryption key via the satellite, which is then stored on the local machine. This key can, however, be used on any machine that has an application that uses the libuplink library.

4.1.2 How data is processed

Through this uplink, a user can upload and download data to and from the network of storage nodes. To better explain how the Storj protocol handles this data, an example file will be used. In

this example we use a file that consists of multiple PDF documents compressed into one file. Storj recommends using files larger than 4 MB due to the fact that storing smaller documents using this protocol wouldn't be economically viable.

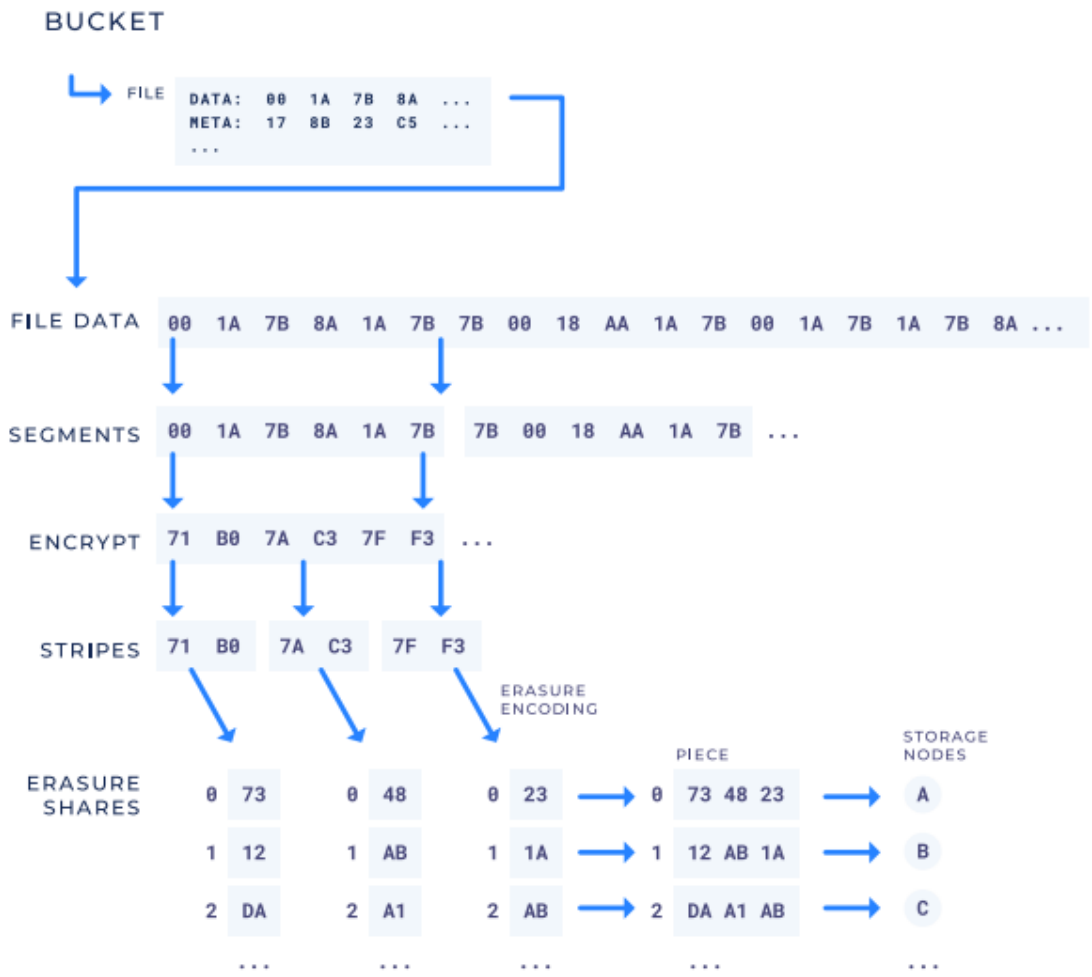

Figure 4--3 The route of a file that is stored on the Storj network (Storj Labs, Inc. 2018, cited 15.04.2019)

Uploading

As seen in figure 4-3, the bucket represents the data of the compressed documents. This data is then segmented by the uplink which encrypts each individual segment with a different nonce. This means that none of the segments are encrypted in the exact same way.

The uplink then sends a request to the satellite, which will first try to identify the requesting uplink. An uplink needs to have an account with a satellite, as mentioned in paragraph 4.1.1. Using API credentials, the satellite can determine that the uplink is authorized for the request. The satellite also determines is the uplink has enough funds to fulfill the request.

If everything is in order, the satellite moves on to select storage nodes. This selection happens based on the requirements of the requester, such as needed bandwidth and geographical location. When the right storage nodes are selected, the satellite sends the contact information of these storage nodes back to the uplink.

At this point the uplink starts to further divide the segment into stripes. The uplink then erasure encodes these stripes of data into erasure shares. These are then combined into a piece. These pieces are then distributed among the selected storage nodes. After each piece is uploaded to a storage node, the hash of this piece is added in the end of the stream. This means that when the data is tampered with, the hash code changes which will serve as a proof whether data is altered or not.

As soon as a complete segment is finished uploading, the uplink uploads a pointer object to the satellite. Within this pointer object the following data is stored:

- Which storage nodes were successful
- What encrypted path was chosen for this segment
- Which erasure code algorithm was used
- The encrypted encryption key
- The hash of each piece's hashes
- Signature

This process is repeated until the last segment is uploaded to the Storj network. A little more information is added to the last segment, namely how many segments there were in total for this file, how large they are in bytes and the starting nonce.

Downloading

When downloading a file, the user again requests this via the uplink. The satellite will determine again if the user is authorized to make this request. When everything is in order, the satellite returns the contact information of the storage nodes that store pieces of the file, and the pointer data.

The uplink will then, with this pointer data, start downloading the file segments in parallel. The downloaded data consist of the erasure shares that were created during the uploading process. These erasure shares can be combined back into the stripes, which can then be decrypted using the encryption key that only the user has.

Deleting

When the user wants to delete files, the uplink sends an request much like it does while uploading or downloading files. The satellite again checks if the uplink is authorized, after which is returns all

the pointer data of the relevant files. With this pointer data the uplink can request the storage nodes to delete all the data that they stored, after which they will return a confirmation that they indeed have deleted all the data as requested.

4.1.3 Security

While developing the Storj network, security and privacy have been the foundation of the protocol. As the whitepaper states: “decentralized storage platforms cannot take many of the same shortcuts data center based approaches can (e.g. firewalls, DMZs, etc.), decentralized storage must be designed from the ground up to support not only end-to end encryption but also enhanced security and privacy at all levels of the system."

Attacks

Storj lists several possible attacks in its whitepaper that cause risks for similar protocols. The attacks it lists are:

- Spartacus attacks
- Sybil attacks
- Eclipse attacks
- Honest Geppetto attacks
- Hostage bytes attacks

Spartacus attack

Spartacus attacks are better known as identity hijacking. In a network of nodes this means that a possible attacker copies the identity of one of the legitimate nodes with the goal of intercepting information. In the Storj network this would mean that a hacker copies the public identity of a storage node so that the attacker can intercept information.

Storj eliminates the possibility of these attacks by using public and private keys. The public key is, as the name already suggests, publicly known, and can therefore be copied by an attacker. However, the Storj protocol requires every message to be signed using the private key. A storage node is the only entity that stores its own private key with which it can sign messages that must be send over the Storj network.

Sybil attack

With this this type of attack a hacker tries to create a large number of nodes with the goal to disrupt the network. This is comparable to the 51% attack, discussed earlier on. When having a large number of nodes in place, the attacker can hijack data, store a lot of data and then suddenly go offline, or disrupt the network by spamming it with messages.

Storj tries to prevent this by implementing the Proof of Work protocol, preventing anyone from easily being able to create a large number of storage nodes all at once. On top of that, storage nodes first have to prove that they are trustworthy, a process that takes relatively long, which makes a sybil attack infeasible.

Eclipse attack

As Storj is a peer-to-peer network, the view that the storage nodes have of the entire network depends on other nodes. Nodes receive this information and distribute it with other new coming nodes. In the case of an eclipse attack, a hacker tries to isolate a node after which it gives the node false information about the network (IoTeX 2018, cited 22.04.2019).

In blockchain networks such as the Bitcoin network this could lead to double spending attacks. In the case of Storj this could mean a hacker can use the storage capacity of a storage node while not paying for it.

Storj prevents these attacks by making sure nodes that just joined the network connect with at least one well-behaving node. These well-behaving nodes are ran by Storj Labs to make sure that every new node has the right information about the network.

Honest Geppetto attack

Storage nodes on the Storj network need to accumulate trust before they can fully participate in offering their storage capacities. It takes time and effort (PoW protocol) and based on certain factors (e.g. uptime) a node is deemed as trustworthy.

During an “honest Geppetto attack”, a hacker runs a big amount of storage nodes for a longer period of time, accumulating the trust of satellites and other nodes. When it has reached a threshold, the attacker suddenly takes the nodes offline with the goal to disrupt the entire network. The main way to prevent such an attack is to have a big network which makes these attacks are ineffective, as mentioned by Storj in its whitepaper.

While the network has a smaller scale, these attacks can be countered by analyzing relatedness to determine who runs the nodes and if they are related, after which an analysis can be made of the chance that a group of nodes is performing a honest Geppetto attack.

Hostage bytes

In the case of a hostage byte attack, the attacker doesn't return data or parts of it. The main goal is to get the owner of the data to pay more money in order to get his or her data from these malicious storage nodes.

Storj prevents the possibility of these attacks by making it possible for the owner to download to needed pieces to reconstruct the data from other nodes. The reconstruction of lost data will be explained in the next paragraph.

A large scale network with many storage nodes makes the success of a hostage byte attack unlikely. On top of that, the data that the attacker takes hostage is of no value to the attacker in the case of the Storj protocol, as it consists of random pieces of encrypted data that are meaningless on their own.

4.1.4 Storj and the CIA triad

Earlier, in chapter 3 Security issues, blockchain was explored from the viewpoint of the CIA triad. It was shown that blockchain performs well on all three aspects of this model, but that the technology still has issues with regards to availability and that in the future quantum computing might cause a problem for the confidentiality of data stored in a blockchain.

As Storj utilizes decentralization and encryption, also this protocol performs well with regards to confidentiality, integrity and availability. However, when it comes to availability a force majeure event may cause issues. When relying solely on Storj, or any similar protocol, data may be inaccessible in the situation of a world wide crisis. Also, in the case of a more local event, such as a power outage on the premises from where one might want to access the data, it will be impossible to access the data due to the reliance on an internet connection. This, however, is also true for any other cloud storage solution.

Finally, when it comes to integrity and confidentiality Storj processes data in a way that it would be hard for a hacker to retrieve data from a storage node. By using peer-to-peer communication in combination with signed messages using private keys, it is nearly impossible to intercept communications between the uplink and a satellite and storage nodes. On top of that, data remains encrypted at all times when stored in the network.

4.1.5 Permanent data storage with Storj

In a network with thousands of storage nodes it is not unlikely that one or more of these nodes don't respond to requests or go offline altogether. Storj aims to avoid inconvenience as a result of nodes going offline by auditing these nodes regularly. These audits are performed by the satellites. They check if a node responds as expected and is indeed storing the data as requested. This way satellites keep track of the reputation of storage nodes.
In case a storage node is marked as bad, the data stored on this node is further distributed in order to prevent the data from getting lost. By auditing regularly, the satellite has an overview of the current degree of distribution of the data. When this distribution falls below a threshold, as a result of nodes going offline and never coming online anymore, the protocol automatically uses the erasure shares stored on other nodes to reconstruct the and redistribute it again.
This way the Storj protocol assures that data is always distributed above a certain threshold. By incentivizing storage nodes Storj encourages storage nodes to behave as expected and to stay online as long as possible.

4.1.6 Costs of Storj

As the v3 version of the Storj network has not been fully launched yet, the costs of storage are based on the costs of the previous network, Storj v2.
Per terabyte stored, Storj charges $15 per month. Aside of that, Storj charges $0,05 per downloaded GB while uploads are free (Leonov, A. 2019. Storj Labs Support, Storj Labs. E-mail message on 28.03.2019).

4.2 Alternative: SIA

In this chapter 4, mainly Storj has been explored as their team of developers provided useful information and responded quickly on questions asked via email. With this information the author of this thesis was able to get a better idea of the protocol without it being necessary to be an expert in blockchain technology.
Storj, however, is not the only distributed storage protocol that is available. SIA is another protocol that facilitates distributed storage. This protocol differs in some areas from the Storj protocol.

The main difference between Storj and SIA is that SIA doesn't rely on a "bridge", which in Storj's case is the satellite. This means that SIA is more decentralized and allows absolute peer-to-peer communication between the user and storage nodes.

By establishing smart contracts, the user that wants to store data and the storage nodes agree on the time that data needs to be stored and other requirements with regards to the needed space etc. When the storage nodes meet all these requirements, the storage nodes get paid in SIA Coin, the cryptocurrency of the SIA protocol.

A downside of SIA is that is has a storage nodes marketplace, meaning that prices change depending on demand. This makes it uncertain what price exactly will be paid in the end. Another downside is that storage nodes can only be paid in SIA Coins, meaning the user is required to own this cryptocurrency to be able to use the storage service (Twilightsparkle 2017, cited 24.04.2019).

5 DISCUSSION

During the research done while writing this thesis, I have learned a lot about how blockchain works in its most basic form. While I have written about this technology before due to my work as a writer for a cryptocurrency news website, I never truly explored how exactly this technology works. I thought it would be important to lay out the pure basics of this technology and include it in this thesis in order to explore the possibilities and risks of this relatively new technology. I do recognize that chapter 4 could've been a thesis topic on its own due to the amount of distributed storage protocols and the available information.

While writing about security issues and blockchain, I realized that most risks revolve around the actual trading of cryptocurrencies through cryptocurrency exchanges or cryptocurrency wallets. The core technology of blockchain itself, however, is designed to be robust by making data stored on a blockchain immutable. Another major advantage blockchain technology has when implemented as a financial network when comparing it to for example banks, is that it offers a high degree of transparency as every single transaction is registered.

In chapter 4 I decided to focus mainly on decentralized storage solutions. Based on the earlier research in chapter 1, it became clear that storing large amounts of data on a blockchain (e.g. adding large amounts of data to blocks and adding them to the chain) is infeasible, let alone storing it for longer periods of time, or even forever. This might also be a reason why we see blockchain mainly being implemented in the context of storing value, such as the Bitcoin network, as this consists of mainly transaction data which is relatively small.

Decentralized storage leverages encryption and decentralization, two very important properties of blockchain technology. Protocols like the Storj protocol offer the possibility to store large amounts of data in a decentralized manner. However, this data is not stored in a blockchain. Storj, for example, encrypts data and distributes it over a large number of storage nodes. This data is encrypted with a key that only the owner of the data has. In the emails that the Storj team sent me, they made clear that it will be possible in the near future to create keys for clients of a company. This way a company could potentially store data of its clients while keeping the encryption key safe in cold storage (on its own premises). When the data needs to be transferred to the client, the company can simply provide them the encryption key with which the client can retrieve the data from the storage nodes that are part of the Storj network. This is technically also possible with SIA, although I haven't truly dug deep into the technology behind this platform.

Potentially, a company can develop a system specially for archiving financial data of its clients, where a protocol such as the Storj protocol serves as the backend. Due to the fact that these protocols are in many cases open source, building a system around them belongs to the possibilities. This way a company can develop its own system that meets all the specific requirements a company has with regards to compliance, the GUI of the application and its usability.

However, in the case that one of the requirements is that data may absolutely not be changed in anyway, decentralized storage in the way Storj and SIA offer it is not an option. As Storj basically mixes bits and bytes, the original document doesn't exist anymore in the original state. Every time a file is downloaded, the data is reconstructed. In other words, it is not the actual original file, but merely a copy of the original file.

Also, when analyzing the Storj network using the CIA triad it became clear that the availability of data stored in the Storj network is good as long as there is an internet connection. When this internet connection falls away, none of the stored data can be accessed.

The commissioning company told me during the first meeting that one of the reasons they'd want to use blockchain storage was so that transferring data would be easier and safer. Protocols such as Storj's make this easy as the data can be accessed from anywhere with the right encryption key. However, if transferring data is the only goal, I do think that there may be cheaper and easier ways to transfer data over the internet where it isn't necessary that data is first distributed before it can be downloaded by a client.

My conclusion based on the research I've done is that decentralized storage absolutely has the potential to fulfill the wishes of a company that needs to archive data in an easy and safe manner and wants to also have the possibility to transfer this data fast and safe to its clients. However, it should not yet be approached as a possible replacement of already existing storage methods and infrastructures as the technology is still relatively young. On top of that, when looking at the Storj protocol from the CIA triad perspective, the availability of the data relies solely on the internet connection of the person or organization storing data in Storj's network. This factor must be strongly considered before a system can be developed around this or any similar protocol.

REFERENCES

Allison, I. 2018. Oxfam in Sri Lanka Will Use Ethereum to Deliver Microinsurance. Hakupäivä 05.03.2019 https://www.coindesk.com/global-charity-oxfam-will-use-ethereum-to-deliver-microinsurance.

Aziz. Guide to Blockchain Scalability: Bitcoin Scalability Problem and Effects. Hakupäivä 30.04.2019 https://masterthecrypto.com/blockchain-scalability-bitcoin-scalability-problem-effects/.

Bajpai, P. 2015. What Is Cold Storage For Bitcoin Hakupäivä 15.03.2019 https://www.investopedia.com/articles/investing/030515/what-cold-storage-bitcoin.asp.

Barton, J. 2019. 17 Major Companies Who Accept Bitcoin [2019 UPDATED]. Hakupäivä 30.04.2019 https://coindiligent.com/who-accepts-bitcoin.

Binance. 2018. What Is a 51% Attack? Hakupäivä 13.03.2019 https://www.binance.vision/security/what-is-a-51-percent-attack.

Bitcoin.it. 2019. Hardware wallet. Hakupäivä 15.03.2019 https://en.bitcoin.it/wiki/Hardware_wallet.

Bitcoin.it. 2019. Private key. Hakupäivä 15.03.2019 https://en.bitcoin.it/wiki/Private_key.

Blockgeeks. 2017. Cryptocurrency Wallet Guide: A Step-By-Step Tutorial. Hakupäivä 15.03.2019 https://blockgeeks.com/guides/cryptocurrency-wallet-guide/.

Blockgeeks. 2019. A Deeper Look at Different Smart Contract Platforms. Hakupäivä 20.03.2019 https://blockgeeks.com/guides/different-smart-contract-platforms/.

Blockgeeks. 2017. What is Ethereum? Hakupäivä 04.03.2019 https://blockgeeks.com/guides/ethereum/.

Blockgenic. 2019. Keeping Your Cryptocurrencies Safe. Hakupäivä 15.03.2019 https://hackernoon.com/keeping-your-cryptocurrencies-safe-14f37e555798.

BusinessDictionary. 2019. Distributed strage. Hakupäivä 08.04.2019 http://www.businessdictionary.com/definition/distributed-storage.html.

Canellis, D. 2018. Hacker exploits EOS smart contract to steal $200K from gambling app. Hakupäivä 20.03.2019 https://thenextweb.com/hardfork/2019/04/29/satoshi-treasure-bitcoin-million-side-quests/.

CCN.com. 2018. EOSBet is the First Crypto-Casino on the EOS Blockchain to Receive Online Gambling License. Hakupäivä 20.03.2019 https://www.ccn.com/eosbet-is-the-first-crypto-casino-on-the-eos-blockchain-to-receive-online-gambling-license.

Choy, W. & Teng, P. 2017. When Smart Contracts are Outsmarted: The Parity Wallet "Freeze" and Software Liability in the Internet of Value. Hakupäivä 20.03.2019, Blockchain and Law.https://www.blockchainandthelaw.com/2017/12/when-smart-contracts-are-outsmarted-the-parity-wallet-freeze-and-software-liability-in-the-internet-of-value/

Coinbase. 2019. Where can I find the private keys for my wallet? Hakupäivä 15.03.2019 https://support.coinbase.com/customer/portal/articles/1526452-where-can-i-find-the-private-keys-for-my-wallet-.

CoinMarketCap. Top 100 Cryptocurrencies by Market Capitalization USD Hakupäivä 30.04.2019 https://coinmarketcap.com/.

CoinMarketCap. 2019. Top 100 Cryptocurrency Exchanges by Trade Volume. Hakupäivä 18.03.2019 https://coinmarketcap.com/rankings/exchanges/.

ComplyAdvantage. 2018. Cryptocurrency Regulations Around The World. Hakupäivä 22.03.2019 https://complyadvantage.com/blog/cryptocurrency-regulations-around-world/.

Cryptocurrencyfacts.com. 2019. What is a Cryptocurrency Exchange? Hakupäivä 18.03.2019 https://cryptocurrencyfacts.com/what-is-a-cryptocurrency-exchange/.

Ethos. 2019. What are Cryptocurrency Exchanges? Hakupäivä 18.03.2019 https://www.ethos.io/what-are-cryptocurrency-exchanges/.

Franco, P. 2015. Understanding Bitcoin. United Kingdom: John Wiley & Sons Ltd.

Frankenfield, J. 2018. Block Header (Cryptocurrency). Hakupäivä 11.03.2019 https://www.investopedia.com/terms/b/block-header-cryptocurrency.asp.

Frankenfield, J. 2017. Nonce. Hakupäivä 11.03.2019 https://www.investopedia.com/terms/n/nonce.asp.

Frankenfield, J. 2018. Paper Wallet. Hakupäivä 15.03.2019 https://www.investopedia.com/terms/p/paper-wallet.asp.

Futurism. 2019. Bitcoin: History and Timeline. Hakupäivä 30.04.2019 https://futurism.com/images/the-entire-history-of-bitcoin-in-a-single-infographic.

Golem. 2019. Golem. Hakupäivä 06.03.2019 https://docs.golem.network/#/.

Good Audience. 2018. Blockchain: how mining works and transactions are processed in seven steps. Hakupäivä 11.03.2019 https://blog.goodaudience.com/how-a-miner-adds-transactions-to-the-blockchain-in-seven-steps-856053271476.

Hohpe, G. & Woolf, B. 2004. Enterprise Integration Styles. Hakupäivä 13.03.2019 http://www.informit.com/articles/article.aspx?p=169483&seqNum=3.

IoTeX. 2018. Eclipse Attacks on Blockchains' Peer-to-Peer Network. Hakupäivä 22.04.2019 https://hackernoon.com/eclipse-attacks-on-blockchains-peer-to-peer-network-26a62f85f11.

Jayachandran, P. 2017. The difference between public and private blockchain. Hakupäivä 07.03.2019 https://www.ibm.com/blogs/blockchain/2017/05/the-difference-between-public-and-private-blockchain/.

Jimi, S. 2018. Blockchain: how a 51% attack works (double spend attack). Hakupäivä 13.03.2019 https://medium.com/coinmonks/what-is-a-51-attack-or-double-spend-attack-aa108db63474.

Jimi, S. 2018. Blockchain: What are nodes and masternodes? Hakupäivä 11.03.2019 https://medium.com/coinmonks/blockchain-what-is-a-node-or-masternode-and-what-does-it-do-4d9a4200938f.

Kharpal, A. 2018. Cryptocurrencies: Regulating the new economy. Hakupäivä 22.03.2019 https://www.cnbc.com/2018/08/09/cryptocurrencies--regulating-the-new--economy.html.

Mougayar, W. & Buterin, V. 2016. The Business Blockchain : Promise, Practice, and Application of the Next Internet Technology. 1. John Wiley & Sons, Incorporated.

Naqvi, S. J. 2017. Converting a Property Rental Paper Contract into a Smart Contract. Hakupäivä 05.03.2019 https://medium.com/@naqvi.jafar91/converting-a-property-rental-paper-contract-into-a-smart-contract-daa054fdf8a7.

Norry, A. 2018. The History of the Mt Gox Hack: Bitcoin's Biggest Heist. Hakupäivä 18.03.2019 https://blockonomi.com/mt-gox-hack/.

Orcutt, M. 2019. Once hailed as unhackable, blockchains are now getting hacked. Hakupäivä 18.03.2019 https://www.technologyreview.com/s/612974/once-hailed-as-unhackable-blockchains-are-now-getting-hacked/.

Ray, J. 2019. A Next-Generation Smart Contract and Decentralized Application Platform. Hakupäivä 04.03.2019 https://github.com/ethereum/wiki/wiki/White-Paper#applications.

Storj Labs, I. 2018. Storj: A Decentralized Cloud Storage Network Framework. Hakupäivä 08.04.2019 https://storj.io/storjv3.pdf.

Tar, A. 2018. Proof-of-Work, Explained. Hakupäivä 28.02.2019 https://cointelegraph.com/explained/proof-of-work-explained.

Thompson, C. 2016. Private Blockchain or Database? Hakupäivä 13.03.2019 https://medium.com/blockchain-review/private-blockchain-or-database-whats-the-difference-523e7d42edc.

Twilightsparkle. 2017. SIA COIN EXPLAINED. Hakupäivä 24.04.2019, Steemit.https://steemit.com/siacoin/@twiligtsparkle/sia-coin-explained

World Crypto Index. 2019. HOW NODES WORK ON THE BLOCKCHAIN. Hakupäivä 11.03.2019 https://www.worldcryptoindex.com/how-nodes-work/

Zmudzinski, A. 2019. Ethereum Co-Founder Vitalik Buterin: ETH Is a Solution to Bitcoin's Limited Functionality. Hakupäivä 04.03.2019 https://cointelegraph.com/news/ethereum-co-founder-vitalik-buterin-eth-is-a-solution-to-bitcoins-limited-functionality.

Azaghal. 2012. Merkle tree. Hakupäivä 07.05.2019 https://en.wikipedia.org/wiki/Merkle_tree#/media/File:Hash_Tree.svg.

Kissel, R. 2013. Glossary of Key Information Security Terms. Gaithersburg, United States: Computer Security Division Information Technology Laboratory. (2), https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

Piscini, E., Dalton, D. & Kehoe, L. 2017. Blockchain & Cyber Security. Let's Discuss. https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-blockchain-and-cyber-security-lets-discuss.pdf

Rush, J. R. 2016. Smart Contracts are Immutable—That's Amazing…and It Sucks. Hakupäivä 07.05.2019 https://medium.com/@tjayrush/smart-contracts-are-immutable-thats-amazing-and-it-sucks-e0fbc7b0ec16.

HD-510 3 question I have

Aleksey Leonov <jira@storjlabs.atlassian.net>
Wo 3-4-2019 15:51
Aan:

tijmen_bult@hotmail.com <tijmen_bult@hotmail.com>

———————
Reply above this line.

Aleksey Leonov commented:

Hello,

We glad to help with your research!

1. The Company can have as much satellites as they want, this is heavy client of the network. It must be online every time, so it's service, which you should keep reliable (to meet your SLA), it's costly to handle it yourself. But you will completely control your data, even metadata.
The alternative is using any of our satellites or satellites from a Tardigrade network (it's brand for reliable and certified satellites with SLA).
2. The uplink is a light client of the network, it will be used to manage buckets, files in your buckets and permissions. So, yes, the company should have as much uplinks, as necessary. This is small application, which can be executed on any PC or even raspberry pi.
3. Files can be any size, just small one will be stored inline in the metadata on satellite in encrypted form, of course. And yes, you can pack them to the big archives if you want. Also, you can configure your uplink to have a more smaller files and treat them as big one with all benefits.
You can encrypt any file with own encryption key. In addition, you will be able to grant a shared access to the file or part of the path ("folder").
4. The Company can build a platform with Storj network as backend to achieve these goals. The satellite(s) will be used as backend service for such platform. You can use uplink CLI, libuplink, S3 gateway with `aws s3` CLI to manage buckets, files and permissions within your platform.
You can generate login/password and encryption keys for your clients, but I think, it's better to allow them doing it themselves. This is question for discussion.
5. The satellite is created to make it impossible: if number of nodes would be lower than a threshold, the recover process will be started and the number of nodes will be normal again. If you mean what happened if all nodes in the world will be down, then, probably there would be a global problem, such as world (or national in case if you limit the geographic region) catastrophe or something like that, in this case we all will have other problems, I think.
At least while satellite is paying to its Storage node Operators for service, nobody will shutdown their nodes.
If you will use your own satellites, then while you pay to your SNO, they will keep your data.
If you will use our or Tardigrade satellites, we will keep a SLA, so, your files will be in safe with this SLA.

With best regards, Alexey
Storj Labs Support

HD-510 3 question I have

Aleksey Leonov <jira@storjlabs.atlassian.net>

Do 28-3-2019 14:33

Aan:

- tijmen_bult@hotmail.com <tijmen_bult@hotmail.com>

———————

Reply above this line.

Aleksey Leonov commented:

Hello,

> You mention satellites and nodes, how are they different?

Satellite This peer class participates in the node discovery system, caches node address information, stores per-object metadata, maintains storage node reputation, aggregates billing data, pays storage nodes, performs audits and repair, and manages authorization and user accounts. Users have accounts on and trust specific Satellites. Any user can run their own Satellite, but we expect many users to elect to avoid the operational complexity and create an account on another Satellite hosted by a trusted third party such as Storj Labs, a friend, group, or workplace.

Storage node This peer class participates in the node discovery system, stores data for others, and gets paid for storage and bandwidth.

see "4.1.1 Actors" section of Whitepaper v3

> When data is stored via these satellites/nodes, is the data encrypted? (I assume yes). If so, how is it encrypted?

Our encryption choice is authenticated encryption, with support for both the AES-GCM cipher and the Salsa20 and Poly1305 combination NaCl calls “Secretbox” [66]. Authenticated encryption is used so that the user can know if anything has tampered with the data.

Data is encrypted in blocks of small batches of stripes, recommended to be 4KB or less [67]. While the same encryption key is used for every encryption batch in a segment, segments may have different encryption keys
Paths are also encrypted.

You and only you have a keys to decrypt data and paths (optionally paths could be not encrypted, on your choice).

See "4.11 Encryption" section of Whitepaper v3

> Can you give me an estimation on how much it would cost per month to store 1TB of data, and does the price per 1TB extra get higher, lower or stay the same?

The price will be fixed near to the production release. The local test v3 is free.
I can tell you only prices for the current (deprecated) v2: $0.015/GB of stored data per month and $0.05/GB of downloads (ingress) per month, the egress (uploads) are free.

> Lastly, where can I find documentation of how the Storj network works in the context of a company that wants to securely store data forever? I was only able to find information mainly focused on the nodes that want to store data on their hard drives.

Your devs (or DevOps) can use the v3 network directly via Uplink CLI (the libuplink is coming soon) or uses any tool (including `aws s3` command) for Amazon S3 to store data via v3 gateway.
You can read how to do it:

- https://github.com/storj/storj/wiki/Test-network
- Uplink CLI tutorial
- S3 gateway tutorial

I suggest you to read a whole Whitepaper v3 to understand how it works in details.

With best regards, Alexey
Storj Labs Support