Cloud Storage Using Block-Chains*

Shahla Atapoor
University of Tartu, Estonia
shahla.atapoor@ut.ee
May 9, 2018

Abstract

These days all of us somehow dealing with centralized cloud storage systems (on our smart phones and personal computers) which are offered by high-tech companies such as Google, Microsoft and Amazon. All these systems are hosted across multiple data centers and when demand increases, the storage capacity of data centers needs to be upgraded which usually are not cheap. From a different point of view, recently due to some security and privacy attacks on centralized storage systems (e.g. SONY Pictures, ICloud) there have been large incentive for decreasing the trust on centralized systems somehow and move to more secure and efficient systems. By appearing blockchain technologies, recently there have been some new ideas on architecture of cloud storages based on blockchain technologies that uses unused storage on personal computers. In this report, after a short review on the standard cloud storages and their disadvantages, we will introduce *Storj* which is a novel blockchain-based decentralized file storage. Additionally, we will get acquaintance with *Metadisk* which is the open source application of Storj that recently is developed and tested widely.

1 Introduction

The concept of cloud storage defines a cloud computing model that data is saved on remote servers accessed from the internet, which is know as *cloud*. A cloud service is maintained and managed by some cloud storage service providers on the storage servers. Companies such as Google, Amazon, Microsoft, Apple and lots of similar high-tech companies are providing cloud storage services. When data is stored in the cloud, it is transferred over TCP/IP from the clients computer to the hosts server in a data center which is the same traditional client-server model. Next, the server copies the data to other servers to comply with industry-standard redundancy policies whereby three copies are made. A graphical representation of current standard cloud storage is shown in Fig. 1.

These cloud services are popular because they offer convenient storage backup that can be accessed anywhere as well as the ability to increase computing resources as needed. However, they come with some drawbacks. In fact the current widely used model of cloud storage that operates through centralized institutions, inherently are insecure in many ways.

Generally, in centralized systems there are variety of threats against personal and sensitive information. On the host servers, data might be thieved, spied, copied, destroyed and so on. So, these services require trust on the part of the user that the service provider will not sell or share their data with other companies or governments. Having data stored and maintained by one central authority requires a lot of faith that a user's data will be given the respect it deserves. Additionally, the prices for such services can be quite expensive because the resources they are offering are built exclusively for the purpose of business.

To overcome the discussed drawbacks on centralized cloud storages, we need a cloud storage model that is not based on trust between client and host. All client private data, including filename, date and other metadata, must be encrypted before any transfer takes place from a clients computer to the cloud. In this case, there can be no centralized point of attack using different attacks. All incentive payments for both resource providers and consumers needs to be automated and made in an anonymous manner, e.g. pseudonymous cryptocurrency.

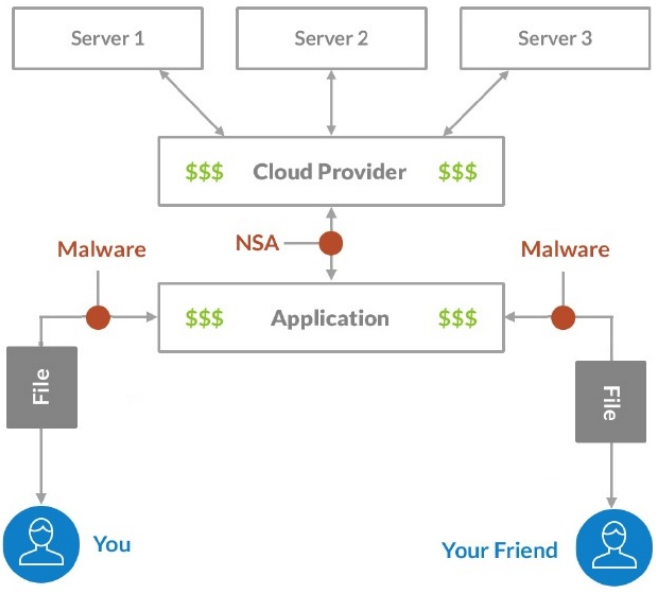

Figure 1: The Standard Model for Cloud Applications [WBBB14].

*This report is prepared as partial fulfillment of the requirements for the course *Distributed Systems Seminar* (MTAT.08.024) in Spring 2018 at Institute of Computer Science, University of Tartu.

Solutions based on decentralized storage network have been known for several years. Projects such as MaidSafe [Mai] and Tornet [Tor] have been sample possible solutions. It is shown that majority of the proposed approaches can not achieve enough security, scalability, and cost efficiency. To deal with the mentioned problems, there have been various endeavors which have gotten to several systems and startups. One of these efficient and novel storage systems is *Storj* which is proposed and released by Wilkinson et al in 2014 [WBBB14].

Storj is a protocol that creates a distributed network for the formation and execution of storage contracts between peers. The Storj protocol enables peers on the network to negotiate contracts, transfer data, verify the integrity and availability of remote data, retrieve data, and pay other nodes. Each peer is an autonomous agent, capable of performing these actions without significant human interaction. Storj gives service using the open source software project *Metadisk* that provides decentralized, cheap, efficient and more secure cloud storage service [WLB14].

This report aims to give an short report on the Storj system and its underlying open-source application Metadisk. To get this end, in the rest of report, after providing some preliminaries related to the topic, we will introduce the mentioned systems and will go through their design and architecture.

2 Preliminaries

2.1 Blockchain and Bitcoin:

During last few years, block-chains and cryptocurrencies are one of the most popular and challenging topics in computer science. Block-chain technology has become a symbol of freedom, transparency and fairness. As the most popular cryptocurrency, Bitcoin has unbelievable features and the most significant one is that creates a secure, reliable digital currency that works based on decentralized system and does not rely on a trusted third party. Bitcoin uses a peer-to-peer network running a proof-of-work algorithm to maintain consensus on the current state of the blockchain. Blockchain is a public digital ledger that records all Bitcoin transactions (transfer of Bitcoin from one set of accounts to another is a transaction). Based on the original paper on Bitcoin, released in 2008 by Satoshi Nakamoto, all electronic cryptocurrencies are based on blockchains which are the entire sequence of all performed transaction and digital signatures [Nak08]. All transactions need to be verified by *miners* before adding to the blockchain. Ownership is transferred when the owner creates a new transaction by signing the hash of both the new owner's public key and the previous transaction, after which this new transaction is added to the chain as shown in the simplified chain of ownership in Figure above (2). Not shown there is that transactions can include multiple

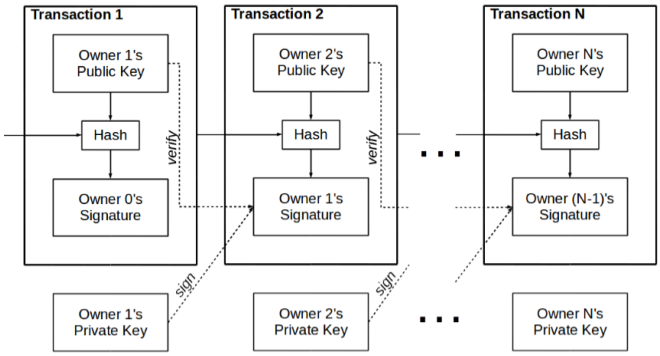

Figure 2: Electronic Coins As A Chain of Digital Signature [Mur16].

inputs, all of which must be digitally signed, as well as multiple outputs. This way, coins do not have to be handled individually (which would be akin to only using the smallest unit of a currency such as pennies), but instead can be combined and split as needed in single transactions. With this coin system of transaction chains it is possible for a recipient to validate the chain of ownership, but they will be unable to prove that the coin has not been spent already. Currency systems using this same method had been around well before Bitcoin and all had run into the same problem [DAI]. The breakthrough for Bitcoin was the development of a new concept that allowed it to not need to rely on a trusted third party like most of the cryptocurrencies that preceded it. This new concept was the blockchain, a public digital ledger that is managed by a peer-to-peer network to maintain consensus on the current state of the system. With this consensus it makes it impossible to double spend a coin because everyone on the network, via the block algorithm they run, has agreed on an exact sequence of transactions that determines the current state of coin ownership.

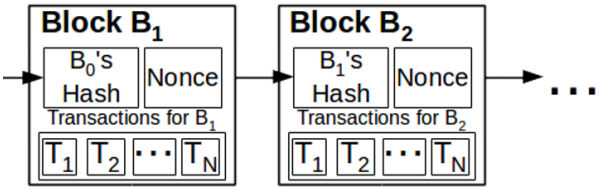

Figure 3: A Simple Blockchain Diagra [Mur16].

It is called the blockchain because at regular intervals a block is added to the existing chain of blocks, with a block being essentially a sequence of the most recent transactions. The block time for Bitcoin averages 10 minutes, meaning the state is changed that often by the addition of a new block. Running the transactions in the blockchain from beginning to end from an initial state (all accounts having zero Bitcoin) will result in the current state of the system. Without getting into too much detail, the decision of which transactions and in which order are

included in the next block is made by a proof-of-work algorithm run by the so called miners running the peer-to-peer network. These miners receive transactions from Bitcoin users and use them to mine the next block. They are rewarded for it with Bitcoin, which is the incentive that maintains a large peer-to-peer network. To mine the next block, miners take the hash of the previous block's hash, all the transactions it has received in sequence, and an integer called a nonce. If the value of the hash is below a certain threshold (which determines the difficulty and thus the average block time) then the mine was successful, otherwise the nonce is increased and the hash is done again. The first miner to achieve this is rewarded and will broadcast the result to the other miners who will validate it, run the transactions in it to get the current state, and then begin work on the next block. Though this algorithm is essentially just busy work at its core, it makes the system secure because the proof-of-work aspect means that an attacker can't just spawn an army of virtual machines to take control. They would need to control more than half of the computational power of the network to control the state of the blockchain. Additionally, it serves to establish a digital currency with no third party oversight by using this busy work to bring about a consensus on the system state.

2.2 AES256-CTR Encryption

The Storj system and application used AES256-CTR to encrypt information before transfer, but they have mentioned that convergent encryption or any other desirable system could be used as well.

The Advanced Encryption Standard (AES) specifies a approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt(decipher) information. A simplified system model of encryption and decryption with block ciphers is shown in Fig. 4. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.

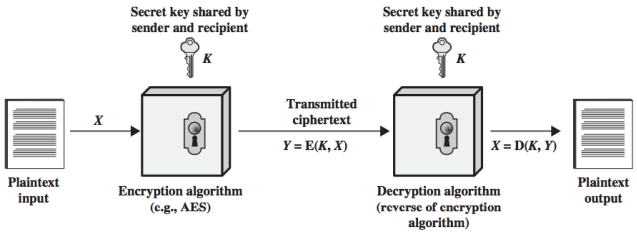

Figure 4: Structure of symmetric encryption [AES].

Block Cipher Modes of Operation. The modes of operation of block ciphers are configuration methods that allow those ciphers to work with large data streams, without the risk of compromising the provided security.

The simplest mode of encryption in cipher blocks is called ECB (Electronic Codebook) Mode, that each plaintext block is encrypted separately. Similarly, each ciphertext block is decrypted separately. A simplified system model of ECB mode is illustrated in Fig. 5.

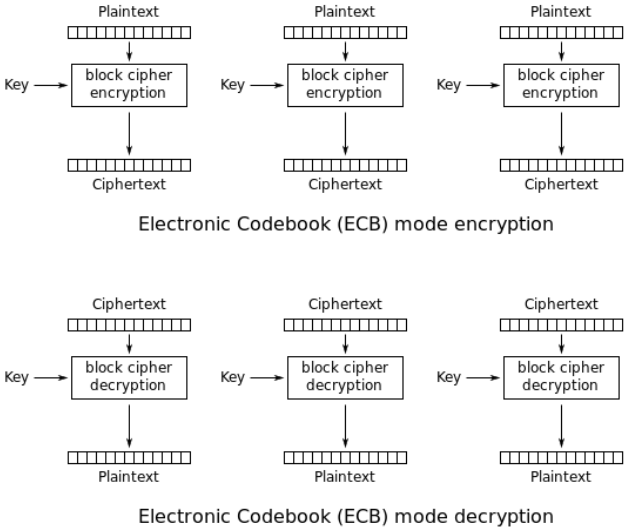

Figure 5: Block cipher with ECB mode [WIK].

The disadvantage of ECB mode is a lack of diffusion. Since ECB encrypts identical plaintext blocks into identical ciphertext blocks, it does not hide data patterns well. In some senses, it does not provide serious message confidentiality, so researcher have proposed different modes which are more secure and are recommended for use in cryptographic protocols.

CTR mode of operation for block ciphers. Among several proposed modes of operation for block ciphers, CTR (counter) mode is one of the most popular ones that both encryption and decryption can be performed using many threads at the same time. The Storj uses implemented AES cryptosystem with key size 256 bits and CTR mode of operation which is introduced below.

Using the CTR mode makes block cipher way of working similar to a stream cipher. keystream bits are created regardless of content of encrypting data blocks. In this mode, subsequent values of an increasing counter are added to a nonce value (the nonce means a number that is unique: number used once) and the results are encrypted as usual. The nonce plays the same role as initialization vectors in modes such as CBC (Cipher-Block Chaining) and CFB (Cipher Feedback) modes which other alternatives [Cry]. The structure of CTR mode is shown in Fig. 5.

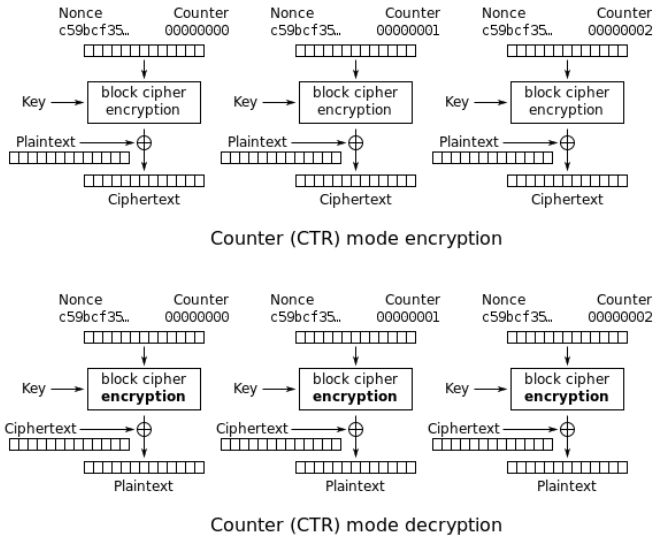

Figure 6: Block cipher with CTR mode [WIK].

3 Storj and Metadisk

In this section we aim to have a short overview to the structure of Storj system and see how it works and what provides for end-users.

3.1 Design

As briefly mentioned in the *introduction*, the Storj is a protocol that creates a distributed network for the formation and execution of storage contracts between peers. Additionally peers can negotiate contracts, transfer data, verify the integrity and availability of remote data, retrieve data, and pay other nodes. Each peer is an autonomous agent, capable of performing these actions without significant human interaction. A graphical representation of the system model of Storj is shown in Fig. 7.

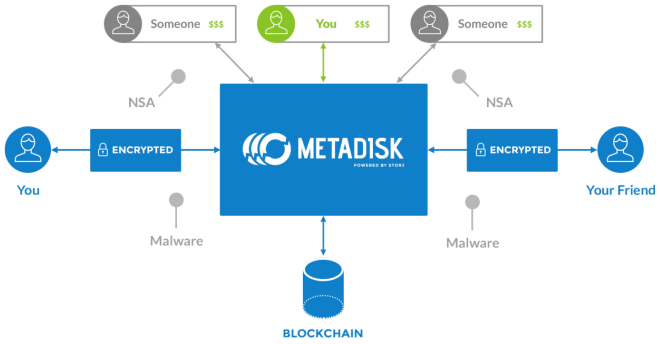

Figure 7: Structure and system model of Storj [WLB14].

3.2 Files as Encrypted Shards

Before being shredded and sending a file to the network, the file should be encrypted in client-side. In order to store an encrypted file on this network, the encrypted file will be divided and each portion of this slices will called *shard*. Security, privacy, performance, and availability are the advantage of sharding. The reference implementation of Storj uses AES256-CTR, but there is the possibility to use the other encryption schemes as well. This protects the content of the data from the storage provider, or farmer, housing the data. The only person who retains complete control over the encryption key, and thus over access to the data is the data owner. In addition, How the data is shredded and where is the location of shards are the information that just the data owner knows. By growing the number of shards it will be exceptionally difficult challenging to locate them without having the knowledge about their previous locations. This implies that security of the file is proportional to the square of the size of the network. There is a recommendation about the choosing the size of the shards which is required in order to preserve privacy the shard sizes should be standardized(8 or 32 MB), although, there is the possibility to negotiate about that. Smaller files may be filled with zeroes or random data. Standardized sizes dissuade side-channel attempts to determine the content of a given shard, and can mask the flow of shards through the network. The advantage of sharding large files such as video is that content and distributing the shards across nodes reduces the impact of content delivery on any given node. Bandwidth demands are distributed more evenly across the network. In addition, the end-user can take advantage of parallel transfer, similar to other peer-to-peer networks. Since peers generally rely on separate hardware and infrastructure, data failure is not correlated. This implies that creating redundant mirrors of shards, or applying a parity scheme across the set of shards is an extremely effective method of securing availability. Availability is proportional to the number of nodes storing the data. Fig. 8 shows the pro-

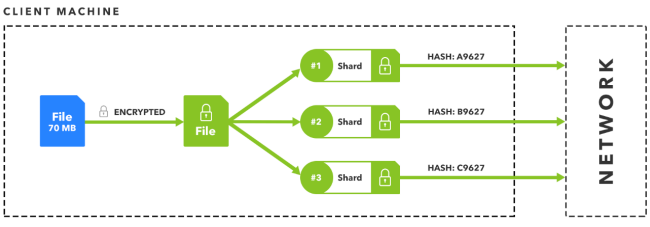

Figure 8: The sharding process [WLB14].

cedure with more details. In summary, the process can be expressed as,

1. Files are encrypted.
2. Encrypted files are split into shards, or multiple files

are combined to form a shard.

3. Audit pre-processing is performed for each shard.

4. Shards is transmitted to the network.

3.3 Redundancy

Cloud object stores typically own or lease servers to store their customers files. They use RAID schemes or a multi-datacenter approach to protect the file from physical or network failure. Because Storj objects exist in a distributed network of untrusted peers, farmers should not be relied upon to employ the same safety measures against data loss as a traditional cloud storage company. Indeed, farmers may simply turn off their node at any time. As such, it is strongly recommended that the data owner implement redundancy schemes to ensure the safety of their file. Because the protocol deals only with contracts for individual shards, many redundancy schemes may be used.

In Metadisk (application of Storj) There are periodic checks availability and non-modifying of the files on the data source which is proved mathematically by Metadisk. In the case that the data source fails, the data can be recovered from another data source. This technique is known as Simple Mirroring. A visualization of this process is shown in Fig. 9.

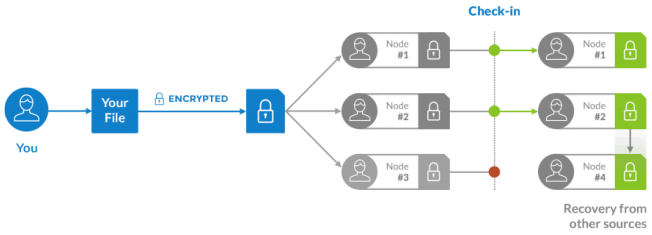

Figure 9: Visualizing the redundancy checks [WBBB14].

3.4 Cost Calculations and Payments

3.4.1 Cost Calculations

As an example test case using the VPS provider Digital Ocean, which could easily host a Metadisk node, one can do some cost estimates. For $5 per month, they are provided with a maximum of 1 TB of transfer [Dig]. This works out to approximately $0.0049 per GB at full utilization. To put this in context, 100 GB would cost us a total of $1.47 to store, with 3 redundancy, and $0.49 to fully retrieve. Dropbox charges $99/year for that same 100 GB of storage. So in essence, under full utilization of the 100 GB for a year, it would cost $1.47 plus extra cost for transfer vs. Dropboxs $99. By adopting a pay per usage model, we avoid the user paying for storage space they dont actually need. As storage media capacity increases at an exponential rate, doubling every 12 months [17], it has become industry-standard practice for cloud storage providers to store files for long periods of time and continually lower their prices per GB to consumers. Therefore, under our full 100 GB usage example, the cost of $1.47 per 100 GB for the first year could easily be $0.74 and approaching zero for the second year and ongoing years. This is competitive with centralized file hosts because even if their cost for storage media halves each year, their (Dropbox's) ongoing operating costs in data center rents, employee salaries, accounting costs, regulatory burden, legal fees, etc. will remain fixed or increase year over year, limiting their ability to compete with a decentralized model that has no such costs. A fast comparison of provided service prices with Dropbox and Meadisk are provided in Fig. 10

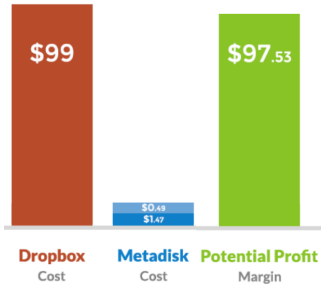

Figure 10: 100 GB of data storage for 1 year, full utilization. Dark blue: storage costs, Light blue: costs for full data retrieval from Metadisk [WLB14].

3.4.2 Payment

Storj is payment agnostic. Neither the protocol nor the contract requires a specific payment system. The current implementation assumes Storjcoin, but many other payment types could be implemented, including BTC, Ether, ACH transfer, or physical transfer of live goats. The reference implementation is going to use Storjcoin micropayment channels, which are currently under development [Bar]. Micropayment channels allow for pairing of payment directly to audit, thus minimizing the amount of trust necessary between farmers and data owners. However, because data storage is Storjcoin allows much more granular payments than other candidate currencies, thereby minimizing trust between parties. In addition, the mechanics of micropayment channels require the total value of the channel to be escrowed for the life of the channel. This decreases currency velocity, and implies that value fluctuations severely impact the economic incentives of micropayment channels. The use of a separate token creates a certain amount of insulation from

outside volatility, and Storjcoins large supply minimizes the impact of token escrow on the market.

New payment strategies must include a currency, a price for the storage, a price for retrieval, and a payment destination. It is strongly advised that new payment strategies consider how data owners prove payment, and farmers verify receipt without human interaction. Micropayment networks, like the Lightning Network [PD16], Implementation details of other payment strategies are left for full version of this report.

4 Conclision

During last few years, due to impressive advantages of distributed systems and blockchain technology, they have found in various applications which we are dealing in our daily life. The *Storj* is one of novel platforms for cloud storage service which properly uses advantages of distributed systems, cryptography and blochchain technology to present a more secure, cheap, and efficient cloud-base storage service. Some of current storage services are known with brands such as *Dropbox*, *Google Drive*, *iCloud* and etc that are presented by high-tech companies and all of the mentioned one have centralized server.

In this report, after reviewing the underlying technologies used in the Storj system, we had a short report on the design of the system and the application that users need to install to use this service. We discussed that the Storj is a protocol that creates a distributed network for the formation and execution of storage contracts between peers. Additionally peers can negotiate contracts, transfer data, verify the integrity and availability of remote data, retrieve data, and pay other nodes. We can share our unused storage in our equipments (such as laptops, personal desktop computers, and so on) and get rewards for this sharing. Naturally, we can rent some storages in this systems for a period of time. Our comparison in the report showed that the Storj system provides much more secure, cheap and efficient service than current standard centralized storage services such as Dropbox.

Acknowledgment: The author has received the IT academy Achievements Stipend for the Spring semester of academic year 2017/2018.

References

[AES] Symmetric cipher model[oneline]. https://notes.shichao.io/cnspp/ch2/.

[Bar] Trustless micropayment channels,[2016]. https://github.com/f483/counterparty-documentation/blob/micropayments/developers/micropayments.md.

[Cry] Crypto IT[oneline]. http://www.crypto-it.net/eng/theory/modes -of-block-ciphers.html.

[DAI] Daily beast[oneline]. https://www.thedailybeast.com/anyone-could-have-invented-bitcoin.

[Dig] Pay-as-you-grow pricing[2014]. https://www.digitalocean.com/pricing.

[Mai] Maidsafe. distributed platform, maidsafe, [2014]. https://www.maidsafe.net/.

[Mur16] Danny Murray. Distributed resource sharing using the blockchain technology ethereum. 2016.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[PD16] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. *draft version 0.5*, 9:14, 2016.

[Tor] Tornet- generic p2p tools, [2013]. https://github.com/bytemaster/tornet.

[WBBB14] Shawn Wilkinson, Tome Boshevski, Josh Brandoff, and Vitalik Buterin. Storj a peer-to-peer cloud storage network. 2014.

[WIK] Block cipher mode of operation[oneline]. https://en.wikipedia.org/wiki/block˙cipher˙mode˙of˙operation.

[WLB14] Shawn Wilkinson, Jim Lowry, and Tome Boshevski. Metadisk a blockchain-based decentralized file storage application. *Technical Report. Technical Report*, 2014.