Distributed File Storage using Blockchain

October 26th, 2017

Atharva Dandekar
adandeka@nyit.edu
ID # 1231011

Nikita Gokhale
ngokhale@nyit.edu
ID # 1208599

Sakhi Saraiya
ssaraiya@nyit.edu
ID # 1232713

Sheharyar Naseer
snaseer@nyit.edu
ID # 1188379

Abstract

As we technologically progress, it is becoming extremely common for both businesses and individuals to use cloud storage services like Google Drive, Dropbox, iCloud and Amazon for storing their files. These services offer them the convenience of storing and retrieving their files from anywhere in the world using the Internet at relatively low costs, but they have introduced their own security and privacy issues because of the way these files storage systems are implemented.

Most file storage services store the users' files without properly securing or encrypting them. Those that do, in most cases, have full read-access to the files internally. This has raised serious privacy concerns, especially during the massive Dropbox hack in 2012 which affected more than 68 million account holders [1], and the iCloud leak in 2014 [2].

In this paper, the authors draw inspiration from the *Blockchain* distributed ledger technology, the *Sia* cryptocurrency and the *Bittorrent* protocol to come up with an alternative strategy for file storage that offers security, privacy and reliability. For implementation, we look towards the powerful BEAM Virtual Machine, offering process supervision, node distribution and a strong Actor model based message-passing system out of the box. The authors design and implement a basic prototype of the suggested strategy, and discuss the results and the challenges of their approach.

Literature Review

This literature review covers three different technologies that this project draws inspiration from; which are the *Blockchain* Distributed Ledger technology, *Sia* (Simple Decentralized Storage) and the *Bittorrent* peer-to-peer file sharing protocol.

Let's start with the Blockchain from where it all started. Going back to 90's the first work on implementing cryptographic secured chain of blocks was described in 1991 by Stuart Haber and W. Scott Stornetta [3]. After about a decade and a half the first distributed blockchain was theorized by an anonymous person or a group called as Satoshi Nakamoto in 2008 and during the following year implemented Bitcoin which then served as a public ledger for all transactions [4].

Blockchain is a relatively simple concept based on the Merkle Tree data structure, but when implemented it leads to a whole new set of technologies. The technology is still in its early stages and there are various ways to implement it which completely depend upon the

problem statement. *"At the superficial level blockchain allows a network of computers to agree at regular intervals on the true state of distributed ledger,"* [5]. What exactly is a distributed ledger? It is a type of database which is shared, replicated, and synchronized among members of a network. Whereas every record in a distributed ledger has a unique timestamp and cryptographic signature attached to it which identifies it uniquely [6]. Such collection or block can contain different types of data varying from account credentials, transactional records or transaction parameters etc. The security of this block is maintained through various cryptographic algorithms and game theory techniques [5].

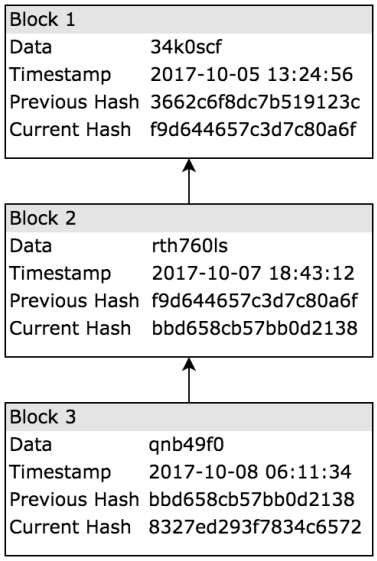

Figure 1 – Example Implementation of Blockchain

Essentially it is a distributed network of all transactions or digital events which take place and are shared among various people who take part it. Transactions which take place in the block are verified by an agreement of majority of the people in the system. The blockchain is immutable which is once the data enters the chain can never be erased [7]. Figure 1 shows an example implementation of Blockchain where each block in the chain, references the cryptographic hash of the previous one which would change drastically if even a small change would be made in the data, timestamp or anything else. In this way, each entry in the Blockchain automatically verifies and secures data in previous blocks, making it virtually impossible to tamper with.

Bitcoin is a peer-to-peer electronic cash system, one of Blockchain's most famous and the very first successful application was brought to life in 2009 by the brilliant Satoshi Nakamoto [8]. The idea behind this invention was to omit the need of a financial institution which would act as intermediate step in online payments sent from one party to another.

Digital signatures posed as the solution though double-spending was a problem which was then solved by introducing a peer-to-peer network where the network timestamps each and every transaction by hashing them into a rolling chain forming a record that cannot be changed without redoing the proof-of-work [9].

The next important work that's been done in the field of decentralized storage is Sia: Simple Decentralized Storage. The idea of Sia was firstly put to implementation at HackMIT 2013 [10]. Sia is basically a decentralized cloud storage platform which competes with existing cloud storage services at various levels. In a world where all the datacenters which are owned by single owners or a company Sia allows anyone to earn something by putting their storage space or hard drive on rent and as far as data integrity is concerned Sia uses redundancy and cryptography to ensure that it is maintained [10].

How Sia works is storage contracts are being made between various peers and these contracts are between the owners of the storage and their clients. What data,

its type and the price of storing it is defined in that contract. There's this requirement where storage owners need to show proof that they are still storing the client's data at regular intervals [11]. Sia uses blockchain to store these contracts and as these contracts and proofs can be verified publically because of its blockchain implementation, there is no need for the clients to personally verify the storage proofs [11].

Sia is the only network where there is complete independence to users to where their data ends up on. More importantly if the owner loses data then they tend to lose the revenue as well as any collateral money that they put up to make sure your data is safe [12].

The protocol or system which enables the reliable and quick transfer of data from many to many peers in distributed network is BitTorrent protocol. BitTorrent is the most widely used peer-to-peer programs ever made. The man behind BitTorrent thought the idea of breaking down files in chunks, encrypting every chunk and storing them in different locations can be used in file sharing. The very first beta version of BitTorrent was released in 2001 [13]. As of 2009 peer-to-peer networks consists of 43% to 70% of all internet traffic [14]. BitTorrent has around 15 to 27 million concurrent users at any time [15].

BitTorrent makes the distribution of files most probably large files which consumes less bandwidth and is easier for the originator. Let us know what exactly BitTorrent protocol is and how it works. It consists of a metafile or called as torrent which contains information or metadata about the host or the publisher, a tracker whose function is to co-ordinate the distribution of files which are shared. A peer should first want to make data available for download must find a tracker for the data then create and distribute the torrent file while other peers can use the information in the torrent file to assist each other in downloading the file [16].

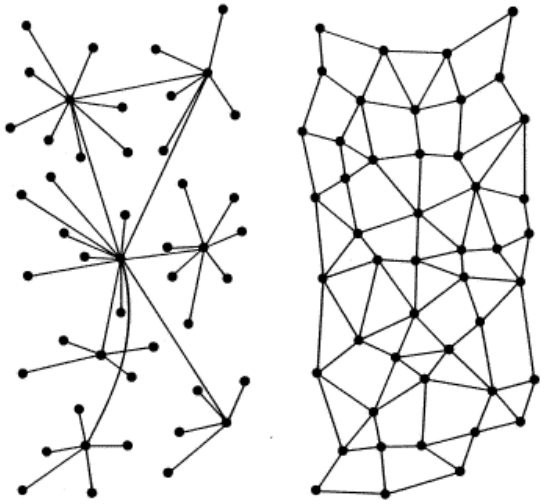

Figure 2 – Decentralized Networks (Left) vs. Distributed Networks (Right) – from "On Distributed Communication Networks" [18]

It is also important that we make a distinction between Decentralized and Distributed networks. The earliest work to discuss this in terms of technological systems is Paul Baran's "On Distributed Communication Networks" [18] from 1964. Figure 2 visualizes their difference very clearly. Bittorrent and IPFS are decentralized, while Blockchain and Bitcoin are distributed and Sia uses a mixture of both (keeping its ledger distributed while keeping the stored data decentralized).

The computer which hosts the original file is called a seed and the file is being broken down in a lot of chunks. The client who wants to download the file need to request the same from a seed using the BitTorrent client. After requesting the client gets one piece of file and over a period of time remaining pieces from various other peers via peer-to-peer communication. Each computer is downloading some pieces of file from some of these peers while simultaneously uploading other pieces of file to other peers.

All the computers which take part in this communication and co-operating in this way are called as *swarm*. How quick this process takes place or the speed of the process depends upon how popular the file is [17].

Proposed System

The authors are implementing a Distributed Blockchain File Storage (DBFS) system which aims on targeting three main concerns of the current world which are security, privacy and reliability.

The system will consist of a basic user interface for the end users to interact with the files they will be uploading to the server. The core of the system is the backend where all the steps will be taken to maintain the three concerns mentioned above. That's where Blockchain comes into play. Our custom implementation of Blockchain would allow us to store data references, its timestamps, executed actions and the cryptographic signatures of the users performing those actions, using Asymmetric key cryptography. This also provides a powerful way to maintain and check the data consistency and keep track of its changes, making it impossible to alter the data history and its integrity. For efficient delivery of data, we'll use Erlang's virtual nodes and message passing by writing a naïve implementation of the *Bittorrent* protocol to reliably transfer the data from many to many peers in a distributed network.

Current Progress

So far, we've created the backend application in the Elixir programming language [19], on top of the Erlang BEAM Virtual Machine. The backend consists of one Supervisor, one GenServer and three modules so far. The three modules are *Block*, *Blockchain* and *Crypto*. The *Blockchain* module provides an interface to work with the Blockchain data structure itself consisting of Linked List of many *Blocks*. The blockchain starts with a zero block, a special data structure to seed the rest of the chain.

The *Block* module defines the block struct, which contains the type of block, creation timestamp, stored data, owner's public key, hash of the previous block, cryptographic signature verifying authenticity of the data and the hash of the entire block. Figure 3 shows this in the form of a native Elixir struct. The cryptographic actions like signing, hashing and verifying blocks are performed using the *Crypto* module which implements RSA Asymmetric key cryptography and the SHA256 hashing function.

```
%DBFS.Block{
  data: %{
    file_name: "Secret Document.pdf",
    encrypted_hash: "596241626D766F2F474F0A36..."
  },
  type:      :file_create,
  timestamp: ~N[2017-10-14 18:33:58.198337],
  owner:     "2D56ADF32B424547494E205055424C9...",
  signature: "8481DA83D6CB75CBE411031AADE5BBD...",
  prev:      "C0579CA7001A896888AF57895FF8021...",
  hash:      "9099520C15740B91FC4BC5B13E1434F..."
}
```

Figure 3 – Our Blockchain Implementation [19]

The *DBFS.Server* GenServer keeps a reference to one Blockchain per node, and provides an interface to other nodes to sync status between them. The Supervisor responsible for making sure that the application recovers gracefully in case of unforeseen issues and errors and that the GenServer is always running.

Our next tasks include implementing our naïve version of Bittorrent to sync files, writing a REST API using Phoenix for interfacing with the Blockchain and finally developing a Web UI that consumes this API and allows the users to securely encrypt and distribute files across all connected nodes in a network.

References

[1] BBC News (2016) – *Dropbox Hacked*

http://www.bbc.com/news/technology-37232635

[2] Wikipedia (2014) – *iCloud leak of photos*

https://en.wikipedia.org/wiki/ICloud_leaks_of_celebrity_photos

[3] Blockchain Whitepaper

https://www.blockchain.com/whitepaper/index.html

[4] The Economist (2015) – *The great chain of being sure about things*

https://www.economist.com/news/briefing/21677228-technology-behind-bitcoin-lets-people-who-do-not-know-or-trust-each-other-build-dependable

[5] MIT Sloan (2017) – *Blockchain explained. MIT Sloan Assistant professor Christian Catalini*

http://mitsloan.mit.edu/newsroom/articles/blockchain-explained/

[6] IBM developerWorks (2016) – *Blockchain basics: Introduction to distributed ledgers by Sloane Brakeville and Bhargav Perepa*

https://www.ibm.com/developerworks/cloud/library/cl-blockchain-basics-intro-bluemix-trs/index.html

[7] Sutardja Center for Entrepreneurship & Technology, Berkeley University of California (2015) – *BlockChain Technology*

http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf

[8] The New Yorker (2011) – *The Crypto-Currency by Joshua Davis*

https://www.newyorker.com/magazine/2011/10/10/the-crypto-currency

[9] Bitcoin.org – *Bitcoin: A Peer-to-Peer Electronic Cash System by Satoshi Nakamoto*

https://bitcoin.org/bitcoin.pdf

[10] Sia – *Sia mission statement*

http://www.sia.tech/about/

[11] Sia: Simple Decentralized Storage – *whitepaper by David Vorick and Luke Champine*

https://www.sia.tech/whitepaper.pdf

[12] News.ycombinator.com – *A cryptocurrency operated file storage network*

https://news.ycombinator.com/item?id=13723722

[13] Department of Telematics, NTNU (2005) – *Peer-to-peer networking with BitTorrent*

http://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf

[14] Wikipedia – *BitTorrent*

https://en.wikipedia.org/wiki/BitTorrent

[15] MLDHT – *BitTorrent mainline DHT measurement*

https://www.cl.cam.ac.uk/~lw525/MLDHT/

[16] Summary of BitTorrent protocol by guest lecturer *Petri Savolainen*

https://www.cs.helsinki.fi/webfm_send/1330

[17] Explainthatstuff (2017) – *BitTorrent*

http://www.explainthatstuff.com/howbittorrentworks.html

[18] Baran, P. (1964) – *On Distributed Communication Networks.* IEEE Transactions on Communication Systems.

https://www.rand.org/content/dam/rand/pubs/papers/2005/P2626.pdf

[19] Naseer, S., et. al (2017) – *DBFS Source Code* on Github.

https://github.com/sheharyarn/dbfs