Blockchain as an IPFS (Interplanetary File System) Storage Index

Tripti Rathee[1], Manoj Malik[2]
[1,2]Department of Information Technology, MSIT, Janakpuri, New-Delhi, India

Abstract- The web has always been an open and neutral place for the world to share knowledge it desires, but its open nature makes it susceptible to many flaws. The current web is insecure; it doesn't offer any security measures by itself. It is centralized and hence poses a danger of single point of failure; this means a central body has total control over it. Blockchain attempts create permanent records of data by distributing and duplicating it and storing it in long immutable ledgers. It provides a democratized trust and validation protocol that has already disrupted banking and is on the verge of overhauling healthcare, financial services, social apps, cryptocurrencies and ICOs. However, the blockchain isn't a complete solution itself to solve the above mentioned problems of the Web. The proof of work consensus mechanisms which are currently in use have slowed transaction speeds to near crippling levels. This makes storing data or large files on the blockchain not feasible. The Interplanetary File System (IPFS) is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. IPFS can be seen as a Distributed Web, and can be utilized as a data store and index the data hash in blockchain. Combined with the latest Asymmetric Encryption protocols the data can be digitally signed and encrypted so it can be only accessed by authorized bodies. This paper aims to combine the advantages of IPFS, Blockchain & Asymmetric Encryption technologies to overcome their shortcomings to create a web that's distributed, secure and immutable.

Keywords – Blockchain, Encryption, Interplanetary File system

I. INTRODUCTION

The Web can be considered as as an archive of human history, it helps in sharing of information and provides various levels and kinds of services. Today's web doesn't enforce any security measures by itself; it is upto the provider to think about the consumer's security. The entire control remains at the provider, this portrait the Centralized nature of the Web. There is a single source of truth on which everyone has to agree, but the source controlled by a central entity may provide no mechanism to verify it. As we consider the Web as a archive of human history, we should also consider to preserve it. The current web provides no guarantee that the content or service one accessed today will be available tomorrow. It is in the nature of web itself that is mutable and hence can change or be removed.

The world has seen a lot of examples where the authority takes over control of the web and restricts access over the usage of it. The web was not mean work like this. The web has be open and available to everyone, this is explained more by the concept of [3] Net Neutrality. HTTP is inefficient and expensive, single source of truth means there's only one node from where the data can be accessed, areas far from the node are susceptible to higher latency rates. HTTP fetches a file from a single computer at a time, instead of getting pieces from multiple computers simultaneously. [4]With video delivery, a P2P approach could save 60% in bandwidth costs. Humanity's history is deleted daily. [5]The average lifespan of a web page is 100 days. It's not good enough of the primary medium of our era to be so fragile. The Internet has been a real accelerator of innovation and one of the great equalizers in human history. But the increasing consolidation of control is a threat to those aspects of the internet. The networks we're using are so 20th Century. It's not enough to organize the world's information. We need to store it such that the world can remember it. [2]IPFS provides deduplication, high performance, and clustered persistence. For Researchers working with, distributing, and analyzing huge datasets, IPFS offers fast performance and [7] decentralized archiving. For the High latency networks are a real barrier of entry to developing world. IPFS provides resilient access to data, independent of connectivity to the backbone or low latency and for Content creators the freedom and independent spirit of the web at full force and at low cost. This can help deliver content in a way which can save considerable money.

1.1 Proposed Solution:

In its entirety, the web is just a very large data store, if this data store was decentralized, immutable in its core and along with it provided a way to verify the truthiness or source of data, this could become a solution that can build a verifiable immutable web. [1]Blockchain and IPFS technologies possess these desired properties; using them together to build a data storage system could become the solution to our problem. Building an information network that will stay up forever is as modern as it gets.

The rest of the paper is organized as follows. The background of blockchain technology and IPFS is explained in section II. The architecture and Requirement specification are presented in section III. Implementation details are given in section IV. Concluding remarks are given in section V.

II. BACKGROUND

2.1 IPFS: Interplanetary File System

The [2] Interplanetary File System (IPFS) is a [9]peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. In some ways, IPFS is similar to the Web, but IPFS could be viewed as a single Bit Torrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model, with content addressed hyperlinks. This forms a generalized Merkle DAG, a data structure upon which one can build versioned file systems, blockchains, and even a Permanent Web. IPFS combines a distributed hash table, an incentivized block exchange, and a self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other.

Instead of using an location address, IPFS uses a representation of the content itself to address the content. This is done using a cryptographic hash on a file and that is used as the address. The hash represents a root object and other objects can be found in its path. Instead of talking to a server, you gain access to this “starting point” of data. This way the system leverages physical proximity. If someone very close to me has what I want, I’ll get it directly from them instead of connecting to a central server. In the lecture example from earlier, the students in the classroom can pull the data from each other without all having to establish their own communication with the a server. With HTTP you are asking what is at a certain location whereas with IPFS you are asking where a certain file is. In order to accomplish this, IPFS synthesizes a few successful ideas from other peer-to-peer systems.

2.2 Blockchain

A [1]blockchain, is a growing list of records, called blocks, which are linked to each other using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and data of the transaction. It is one type of a distributed ledger that consists of replicated, shared, and synchronized data over the Internet. The blockchain needs no centralized control actor or centralized data storage for maintaining its data. The first introduction of a blockchain was by Satoshi Nakamoto in 2008 and implemented as a core part of Bitcoin. There are various blockchains with different goals (e.g., Bitcoin uses its own blockchain called Bitcoin blockchain [1], whereas Ethereum uses its own blockchain called Ethereum blockchain [13]) but the followings are common elements:

Replicated ledger: The history of all transactions among nodes in a blockchain are securely stored. A block consists of transactions that are append-only with immutable past. The blocks are distributed and replicated among the blockchain nodes.

Cryptography: Integrity of all transactions shared among the blockchain nodes is supported with digital signatures and specialized data structures (e.g., hash based data structure called Merkle tree [14]). Authenticity of transactions is supported with digital signatures. Privacy of transactions is also supported with anonymous addresses for transactions.

Consensus: Transactions that are exchanged among the blockchain nodes over the Internet need to be validated before adding to the existing blocks. A consensus among the blockchain nodes is required for the validation. For a public blockchain, a representative consensus algorithm is Proof-of-Work (PoW) [6], which is used by Bitcoin. Practical Byzantine Fault Tolerance (PBFT) [15], [16] is a representative consensus algorithm used by HyperLedger Fabric [17] for a private blockchain.

Peer-to-Peer networking: All transactions are shared without a centralized control actor over the Internet. In other words, the blockchain nodes are connected through a peer-to-peer network over the Internet, not through the client-server model, due to no trust entity involvement.

By design, a blockchain is resistant to modification of the data. It is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority. Although blockchain records are not unalterable, blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been claimed with a blockchain.

Blocks hold batches of valid transactions that are hashed and encoded into a Merkle tree. Each block includes the cryptographic hash of the prior block in the blockchain, linking the two. The linked blocks form a chain. This iterative process confirms the integrity of the previous block, all the way back to the original genesis block.The block time is the average time it takes for the network to generate one extra block in the blockchain. Some blockchains create a new block as frequently as every five seconds. By the time of block completion, the included data becomes verifiable. In cryptocurrency, this is practically when the transaction takes place, so a shorter block time means faster transactions.

By storing data across its peer-to-peer network, the blockchain eliminates a number of risks that come with data being held centrally. The decentralized blockchain may use ad-hoc message passing and distributed networking.

Peer-to-peer blockchain networks lack centralized points of vulnerability that computer crackers can exploit; likewise, it has no central point of failure. Blockchain security methods include the use of public-key cryptography. A [10] public key is an address on the blockchain. Value tokens sent across the network are recorded as belonging to that address. A private key is like a password that gives its owner access to their digital assets or the means to otherwise interact with the various capabilities that blockchains now support. Data stored on the blockchain is generally considered incorruptible.

Every node in a decentralized system has a copy of the blockchain. Data quality is maintained by massive database eplication and computational trust. No centralized official copy exists and no user is trusted more than any other. Transactions are broadcast to the network using software. Messages are delivered on a best-effort basis. Mining nodes validate transactions, add them to the block they are building, and then broadcast the completed block to other nodes. Blockchains use various time-stamping schemes, such as proof-of-work, to serialize changes. Alternative consensus methods include proof-of-stake. Growth of a decentralized blockchain is accompanied by the risk of centralization because the computer.

Blockchain technology also has some downsides. The blockchain ensures a strong degree of security for its chain (i.e., a set of blocks linked) but the risk of managing private keys exists. The private keys are used to prove ownership of a certain asset or data in the blockchain, but those keys could be lost or stolen by attackers. Another issue is scalability. As the blockchain is immutable and append-only, it must maintain a continuously growing list of blocks. For the Bitcoin blockchain, its size reached 100 GB on December 16, 2016 and continues to grow. Also there is an issue regarding the block size, e.g., how many transactions are included in one block. Network performance is also considered. The transactions per second are one of major performance factors that most of the blockchain implementations is trying to improve.

As transactions need to be broadcasted to blockchain nodes connected through a peer-to-peer network, the network could be easily congested. Network congestion would be a critical issue with the growth in transactions and the limited block size.

2.3 Blockchain+IPFS

IPFS and the Blockchain is a perfect match. One can address large amounts of data with IPFS, and place the immutable, permanent IPFS links into a blockchain transaction. This timestamps and secures the content, without having to put the data on the chain itself.

III. PROPSED ARCHITECTURE AND REQUIREMENT SPECIFICATION

3.1 Architecture

A common blockchain stores the complete transaction details, the data, timestamps etc. in the chain itself, while this data becomes immutable and replicated across multi nodes, this size of data becomes a serious source of problems. This current way of data store solves half of the problem but also introduces more. [6]Storing entire data instructions increases the byte size of the chain and terribly affects the blockchain performance. It decreases down replication speed and increases cost of computation. See Figure 3.1

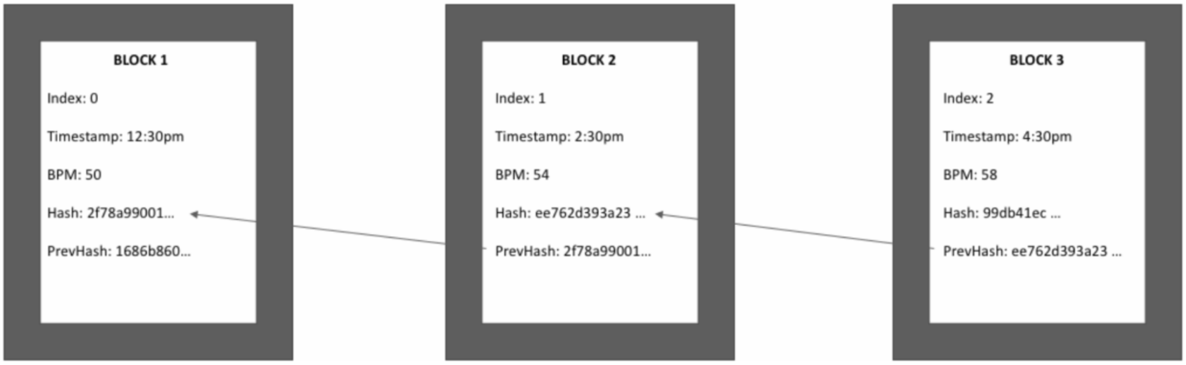


Fig 3.1 Common Blockchain Data Structure

3.2 Proposed Blockchain Data Store

Instead of storing the entire data, the data can be stored at a different location and an address pointer to that location with the hash of the data can be stored in the chain. See Fig 3.2 This adds the following advantages:

Byte size of the chain is kept under control and to it minimum. The address pointer and data hash require very less bytes than the data itself. A lightweight chain means that it can be quickly replicated across the network and consensus algorithms can easily work with them.

Storing the hash of the data enables immutability by giving data verification advantage from the file hash.

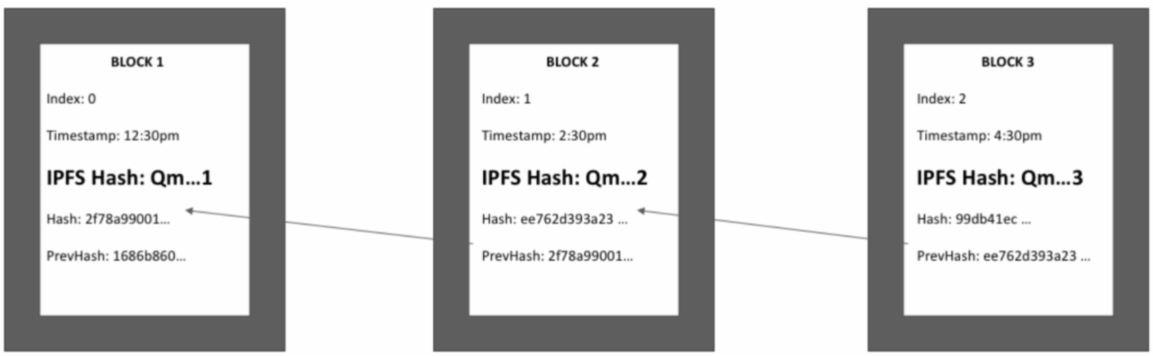


Fig 3.2 Proposed Structure of Blockchain storing data hash instead of complete data itself

This architecture can be divided into two parts, the first one deal with storage of data in the IPFS network and indexing it in the Blockchain while the second part concerns with the retrieval of data from IPFS using the indexed hash from blockchain.

3.3 Specifications

The basic working implementation of this requires a blockchain and an IPFS daemon working on the network. A simple blockchain with in-memory storage, a simple proof of work algorithm and a consensus algorithm to resolve conflicts is required. The blockchain is interfaced over the network using a python server framework called FLASK. The IPFS daemon can be installed on nodes and the daemon can be started to interface with it.

Blockchain Specifications

Transactions: A transaction contains the file hash and any metadata about the files, like the author name, created at, updated at timestamps etc.

Proof of Work: A simple proof of algorithm: Find a number that when hashed with the previous block’s solution, a hash with 4 leading 0s is produced.

Consensus: Consensus decision-making is the group decision-making process in which group members develop, and agree to support a decision in the best interest of the whole. A conflict is when one node has a different chain to another node. To resolve this, the rule is that the longest valid chain is authoritative. In other words, the longest chain on the network is the de-facto one. Using this algorithm, we get the Consensus amongst the nodes in our network.

Consensus Algorithm: “Longest verifiable chain is the authoritative chain.”

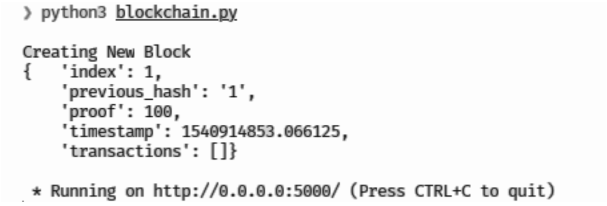
Decentralized: Multiple servers can be spun up on different ports of a same node or on different nodes on the same network. Each node has to register itself with the other node with the nodes/register’ api.

Block: Object representation of a block:

```
block: {
 'index': 1,
  'timestamp': 1506057125.900785,
  'transactions': [
     {
       ‘data’: "file__hash",
       ‘name’: "file__name",
      “author’: "file__author",
     }
  ],
  'proof': 324984774000,
  'previous_hash': "2cf24dba5fe1b17425e73043362938b9824"
}
```

Blockchain Application Programming Interface

The Blockchain Class and methods have been exposed to the network using the [11] Flask python micro framework Flask server exposing the Blockchain

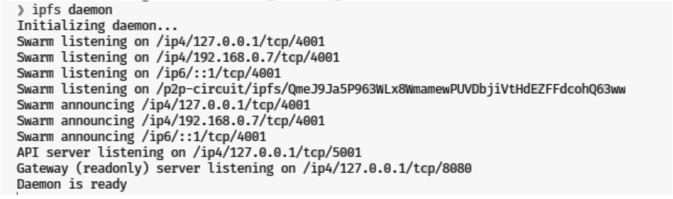


```
❯ python3 blockchain.py

Creating New Block
{   'index': 1,
    'previous_hash': '1',
    'proof': 100,
    'timestamp': 1540914853.066125,
    'transactions': []}

 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Figure3.3. Blockchain server & node running

IPFS Daemon

The [12]IPFS daemon is started on all nodes that require upload or download of data



```
❯ ipfs daemon
Initializing daemon...
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/192.168.0.7/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit/ipfs/QmeJ9Ja5P963WLx8WmamewPUVDbjiVtHdEZFFdcohQ63ww
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/192.168.0.7/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

Figure3.4: IPFS Daemon service running

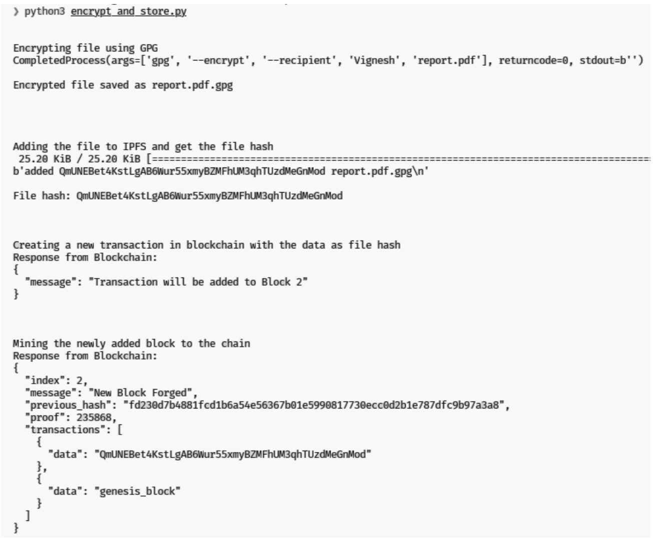
IV. IMPLEMENTATION DETAILS

4.1 Indexing And Storing Data:

File is Encrypted using Public Key: File data can be optionally secured by the author using a appropriable encryption system. Since the stored files are open, if the author wants to restrict access to it content, it has to be encrypted

Result file is hashed: IPFS produces a SHA-256 hash of the file. The file is stored in the network and this hash is returned. This unique hash is used to identify and retrieve the file from the network.

Returned file hash is saved in a transaction in the blockchain, a block is created periodically which preserves the contents of the chain.



```
❯ python3 encrypt_and_store.py


Encrypting file using GPG
CompletedProcess(args=['gpg', '--encrypt', '--recipient', 'Vignesh', 'report.pdf'], returncode=0, stdout=b'')

Encrypted file saved as report.pdf.gpg




Adding the file to IPFS and get the file hash
 25.20 KiB / 25.20 KiB [==================================================================================================
b'added QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod report.pdf.gpg\n'

File hash: QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod



Creating a new transaction in blockchain with the data as file hash
Response from Blockchain:
{
  "message": "Transaction will be added to Block 2"
}



Mining the newly added block to the chain
Response from Blockchain:
{
  "index": 2,
  "message": "New Block Forged",
  "previous_hash": "fd230d7b4881fcd1b6a54e56367b01e5990817730ecc0d2b1e787dfc9b97a3a8",
  "proof": 235868,
  "transactions": [
    {
      "data": "QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod"
    },
    {
      "data": "genesis_block"
    }
  ]
}
```

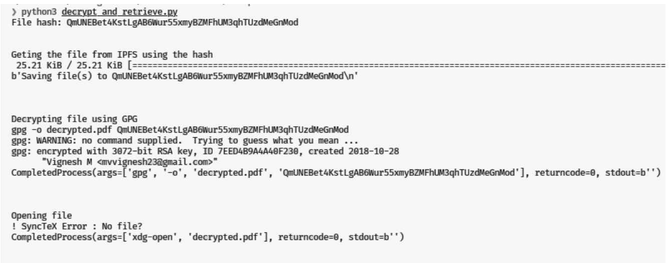
Figure 4.1: Encryption, Upload & Indexing

4.2 Retrieving Indexed Hash And Data:

Get file hash from Blockchain: To identify and retrieve file from IPFS we require the file hash. File hash is grabbed from the blockchain.

Get encrypted file from IPFS using file hash: A GET issue for the file with the file hash can be requested to the IPFS network which will return the file. The name of the file will be its hash.

(Optional) Decrypt file using Private key: If the file was initially encrypted before saving, it has to be decrypted first before it is ready for use.



```
❯ python3 decrypt and retrieve.py
File hash: QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod


Geting the file from IPFS using the hash
 25.21 KiB / 25.21 KiB [=========================================================================================
b'Saving file(s) to QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod\n'



Decrypting file using GPG
gpg -o decrypted.pdf QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod
gpg: WARNING: no command supplied.  Trying to guess what you mean ...
gpg: encrypted with 3072-bit RSA key, ID 7EED4B9A4A40F230, created 2018-10-28
      "Vignesh M <mvvignesh23@gmail.com>"
CompletedProcess(args=['gpg', '-o', 'decrypted.pdf', 'QmUNEBet4KstLgAB6Wur55xmyBZMFhUM3qhTUzdMeGnMod'], returncode=0, stdout=b'')



Opening file
! SyncTeX Error : No file?
CompletedProcess(args=['xdg-open', 'decrypted.pdf'], returncode=0, stdout=b'')
```

Figure 4.2: Retrieve, Download & Decrypt

V. CONCLUSION

The drawback of the Blockchain that it can't hold large sized data, large sized data slows the cloning and storing process and hence slows the entire network. This has been fixed by introducing IPFS storage to store the data and pass on the immutable urls pointing to the data to the chain. The files are encrypted before storage and decrypted on retrieval by an authentic user.

Each procedure of the proposal has been described in details. In addition, one practical example showing how the proposed method works has been presented. As discussed, the proposed method has a room for improvement.

VI. REFERENCE

[1] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008
[2] Juan Benet, "IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3)", 2014
[3] Shane Greenstein, Martin Peitz, Tommaso Valletti, "Net Neutrality: A fast lane to understanding the trade-offs", National Bureau of economic research, January 2016
[4] Kien Nguyen, Thinh Nguyen, "A P2P VIDEO DELIVERY NETWORK (P2P-VDN)", 2009, pp. 1-5.
[5] Mike Ashenfelder, "The Average Lifespan of a Webpage", https://blogs.loc.gov/thesignal/2011/11/the-average-lifespan-of-a-webpage/, 2011
[6] Hyperledger, "Hyperledger Blockchain Performance Metrics", V1.01, October 2018.
[7] Protocol Labs, "Filecoin: A Decentralized Storage Network" July 19, 2017
[8] CGI Group Inc, "Public Key Encryption and Digital Signature: How do they work?", 2004
[9] Antonio Tenorio-Fornés, Samer Hassan and Juan Pavón, "Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent Oriented Framework", June 15, 2018
[10] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.
[11] Armin Ronacher, "Flask Docs", http://flask.pocoo.org/docs/"
[12] Juan Benet, "IPFS Documentation", https://docs.ipfs.io/
[13] G. Wood, ''Ethereum: A secure decentralised generalized transaction ledger,'' Tech. Rep., 2014.
[14] R. C. Merkle, ''A digital signature based on a conventional encryption function,'' in Proc. Conf. Theory Appl. Cryptogr. Techn., 1987, pp. 369–378.
[15] M. Castro and B. Liskov, ''Practical byzantine fault tolerance and proactive recovery,'' ACM Trans. Comput. Syst., vol. 20, no. 4, pp. 398–461, Nov. 2002.
[16] A. Clement, E. Wong, L. Alvisi, M. Dahlin, and M. Marchetti, ''Making byzantine fault tolerant systems tolerate byzantine faults,'' in Proc. 6th USENIX Symp. Netw. Syst. Design Implement., 2009, pp. 153–168.
[17] C. Cachin, ''Architecture of the hyperledger blockchain fabric,'' Tech. Rep., Jul. 2016.