International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 3

Avoiding Data Piracy in Artworks using Blockchain

Arun Kumar G, Hariharan A, Ms D Sasikala[#]

[#]Assistant Professor, B. E, Computer Science and Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, Tamil Nadu, India

ABSTRACT

With the tremendous development of internet, we can share any media from anywhere in the world. This paved the way for data sharing illegally – pirated versions of data shared among persons. In our proposed system, the original data (use case taken – Song) will be embedded with a hash value (SHA-256) and then deployed to Inter Planetary File System (IPFS) and shared through Ethereum Blockchain, enabling deployed data unaltered. The IPFS returns a 46-bit length hash for each of the file being upload. Usage of Ethereum blockchain ensures each and every transaction cryptographically hashed and logged. Also, the data deployed on IPFS sharable but with hidden hash identity for each of it, which is not known by the person who gets that data. The owner of the original data thus shares the data and will be having the log maintaining hash identity for each of the person to whom the data owner shares. The IPFS storage enables Peer to Peer (P2P) data transfer through decentralized network. The person will be given the IPFS hash to download the data. When the data if pirated is known to the owner, he computes the hash value from the pirated version and identifies the person pirated that data and avoids sharing data to that person next time

Keywords: *data piracy, blockchain technologies, artworks, decentralized storage, data sharing*

INTRODUCTION

Blockchain is the recent trend in the technology with a tremendous interest from finance [1] to utilities [2], from medical to grid technologies. The reason for this greater interest is, with blockchain technology we can achieve the same amount of functionality and certainty without the need of any trusted third-party. This breaks the limitations of traditional model of networking by replacing it with trust less networks. A decentralized network works without any server, treating each and every participating node with equal accesses. Even through the network parties won't trust each other they can still communicate with each other in a trust less environment through Blockchain. Blockchain is a type of distributed ledger technology which contains information about transactions or events. Each block consists of transactions information which are guaranteed by cryptographic hash functions and cannot be tampered or forged. A cryptographic hash function is used to produce a specific hash value for the each of the transactions that takes place. For instance, Bitcoin uses SHA-256 hash [3], Ethereum [3] uses Ethash to produce hashes. Ethereum Blockchain is an open source Blockchain platform for developing Dapp [10]. Ethereum uses Smart Contract functionality to digitally assess contracts among people. It automatically takes care of the changes and re-deploys in to the ethereum network whenever any modifications, change occurs to the smart contract code. Ethereum can be developed and tested both locally and in the Main Ethereum Network. Data transfer when taken place without the permission of the owner of the data makes it illegal which increases the data piracy. This not only increases data piracy but also incurs loss to the data owner who is selling that data. Thus, there is a need for secure, reliable medium is necessary to enable data sharing legally.

Motivated by the Blockchain technology, we propose a new system that can be used to avoid data piracy by secure and legal transfer of data from one to another. In our proposed system we have taken use case –

Song to transfer it through decentralized network, by embedding a secret hash identify for each of the person having that song. The hash identity will be unique for each of the user having that data. This hash identity shall be used to identify, to whom the data really belongs to. Further, our paper demonstrates how the secret data embedded into the data and then deployed into the IPFS network and then shared using Ethereum Blockchain. The paper is sectioned into: Section 2 explains some of the Related works, Section 3 explains an overview of Ethereum Blockchain, Section 4 with our Proposed system with various modules, Section 5 has IPFS deployment, Section 6 with ethereum usage, finally Section 7 with overall dataflow of the system. And at last we have conclusion and list of references.

2. RELATED WORKS

In this section we explore some of the related work to our proposed scheme.

2.1 Reduced Share Size Audio Secret Sharing

This work [4] defines secret sharing of messages by means of splitting the original message into some predefined number of times. The author Sonali Patel, Tejal Chavan, P.R. Deshmukh proposed a Matrix Projection technique which can be used to separate the original message into number of shares. To formulate the original secret, all the shares made are needed necessarily. Anything less than the specification provided would render the message in an unattainable form. The proposed system holds a threshold (t, n) audio secret sharing scheme based upon the matrix projection. Among 'n' shares of the original message, a minimum of 't' shares are required for the reconstruction of the original message. If there are less than 't' shares the audio file will not be revealed which implies a Perfect Secret Scheme (PSS).

The sender initially splits the original secret audio file into 'n' shares and then shares to the receiver via multiple routing paths. At the receiver side the reconstruction of the original audio file takes places from 't' shares among the 'n' shares available. This can enforce security in sharing secret audio among the network. However, the routing paths and the network stability plays a crucial role here, and the shares are distributed without any cover medium, which decreases its integrity.

2.2 Blockchains Everywhere - A Use Case of Blockchains in the Pharma Supply Chain

The work here presented is a start-up [5] modem.io which uses IoT (Internet of Things) sensor devices leveraging block chain technology to assert data immutability and public accessibility of temperature records, while reducing operational cost in pharmaceutical supply-chain. The Authors Thomas Bocek, Bruno B. Rodrigues proposed that sensor devices which are attached to the shipped parcels monitors each of the parcel during the shipment to fully ensure GDP regulations. The data collected from these sensor devices (such as temperature record, location, package status) were analysed and transferred to the Blockchain. The Blockchain used here is Ethereum which uses smart contract that assesses against the product attributes. A smart contract is a contract that can verify its correctness and enforce some predefined rules automatically, thus smart contracts are self-executing and self-enforcing. The benefit of using smart contract is that these can be evaluated automatically. The Ethereum, which is a tamper-proof fully decentralized system that can be used at a low cost, ensuring each data on the Blockchain deployed unmodifiable.

Modum.io is used to monitor all the necessary data during the shipment of the parcel by means of sensing their temperature, heat, position, etc with IoT sensor devices. The collected data are then analysed against the smart contracts by the Ethereum Virtual Machine (EVM). The EVM compiles the smart contracts and executes them to check regulations of the medical package. This technology guarantees data integrity and makes it impossible to tamper with records.

2.3.An improved P2P File System scheme based on IPFS and Blockchain.

An enhanced P2P file sharing system is proposed by Yongle Chen, Hui Li, Kejiao Li and Jiyang Zhang. With IPFS characteristics and its high throughput a new model [6] is suggested that provides a novel zigzag-based storage methodology, to improve the block storage model that the IPFS provides. Lots of data gets transfers everywhere and it is quite difficult to make version control over them. IPFS does not take into account the special circumstances of large content service providers. It is not easy for them to join the IPFS network. They need a set of nodes to store large amounts of data, and they need a new strategy to ensure the availability and reliability of

large amounts of data and also reduce storage overhead. Since the block storage model that IPFS provides can bring some loss of benefits to service providers in the event of nodes failure, a new method is proposed that combines replication scheme and erasure codes scheme.

The proposed idea involves improvement in the high throughput problem of IPFS and provides a new solution for the content service providers (CSPs) to better participate in the network scheme. The Blockstack layers idea is used here which has four layers: 1: Blockchain Layer, 2: Virtual chain Layer, 3: Routing Layer, 4: Storage Layer with two layers in control plane and two layers in data plane and also enabling the high throughput with better participation of the CSP.

2.4.Enabling Localized Peer-to-Peer Electricity trading among Plug-in Hybrid Electric Vehicles Using Consortium Blockchain

The authors Jiawen Kang, Rong Yue, Xumin Huang, Sabita Maharajan proposes a localized Peer-to-Peer (P2P) electricity trading model [7] for locally buying and selling electricity among Plug-in Hybrid Electric Vehicles (PHEVs) in smart grids. Traditionally we have a transport system that does transport of electricity over large and complex system. The proposed model not only solves complex electricity transport problem but also achieves demand response by providing incentives for discharging PHEVs to balance local electricity demand out of their own self-interests. Since transaction security and privacy protection issues are there, they explored a promising consortium blockchain technology to improve transaction security without reliance on a trusted third party. The proposed system provides a smart grid environment in which trust less parties can also participate, and charge, discharge their vehicles by implementing blockchain protocols. They used electricity pricing for each of the charging and discharging PHEVs and logged these data cryptographically into the Blockchain. The amount of traded electricity among PHEVs (including charging and discharging) are solved by an iterative double auction mechanism to maximize social welfare in this electricity trading. The Iterative Double Auction Algorithm implements a scheme in which initially the charging PHEVs and the discharging PHEVs submit their bid prices to the auctioneer. The auctioneer then broadcasts a new allocation for the charging and discharging PHEVs. In addition, auctioneer monitors localized P2P electricity trading in real time. The Localized Peer-to-Peer Electricity Trading System with Consortium Blockchain has Local Aggregators (LA) which has a transaction server, an account pool, and a memory pool. During the localized P2P electricity trading, the PHEV first choose their own roles according to electricity demand and energy states. The double auction mechanism is achieved among the charging PHEV, discharging PHEV, and an Auctioneer within the local aggregator.

2.5.Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications

A secure and user-centric convenient healthcare sharing system [8] is proposed in this work by Xueping Liang, Juan Zhao, Sachin Shetty, Jihong Liu, Danyi Li. Since there is a huge risk in the potential privacy issues and vulnerabilities existing in current personal health data storage systems, the usage of blockchain is proposed here that can ensures greater privacy. To preserve the integrity of the data collected from the personal wearables, the health monitoring data is collected and then hosted on the cloud. The cloud data then anchored to the Blockchain. A mobile application is deployed to collect the health data from personal wearable devices and then analysed. The health data collected then will be synchronized with the cloud for data sharing with healthcare providers and health insurance companies. With tree-based data processing and batching methods, handling large sets of personal health data collected makes it more scalable and also achieves greater performance.

The usage of blockchain here has three functionalities: data collected from both wearables devices and health-care providers, each of the hashed entry is uploaded to the blockchain network for integrity protection. For the healthcare provider and the health insurance company to access the personal health care data, they need to get the permission from the data owner.. Since the health data comes from across variety of devices all day, it results in a large data set. To adopt this large data set collection Merkle tree is adopted which has stability requirement and most importantly improves the efficiency to validate the integrity of the data. Merkle tree is a binary tree structure where the input is a list of hashed data records. Every two nodes are grouped and then treated as a leaf for the Merkle tree. This is repeated until

there is a single hash which will become a tree root, that is, the Merkle root. With Blockchain each request and update from healthcare providers are recorded and anchored to the blockchain network, making actions towards healthcare data accountable.

3. ETHEREUM BLOCKCHAIN OVERVIEW

Ethereum is an open-source, public blockchain based distributed computing platform. It uses smart contract (scripting) functionality. It supports a consensus mechanism over a decentralized network by the help of the smart contracts. A smart contract is a contract that is digitally facilitate, verify, or enforce the performance of a contract. Smart contracts allow the performance of the transactions without any trusted third parties. The smart contracts are compiled down to EVM [9] bytecode and deployed to the Ethereum blockchain for execution. The smart contracts are written in solidity language which is a contract oriented high-level programming language. It is a statically typed programming language designed and developed for smart contract execution. All the transactions information is cryptographically hashed, and irreversible. Ethereum uses Ether [9] – which is a cryptocurrency whose blockchain is generated by the Ethereum platform. Ethers can be transferred among the Ethereum network from one node to another and it is used to compensate the participating mining nodes for computations performed. The Ethereum nodes does the transaction and gets approved only when all the mining nodes in the network approves the transaction by proof-of-work. The nodes in the network share a consensus mechanism and works for every transaction approval. This ensures secure and valid transactions in the trust-less network and also avoids double-spending. Ethereum has a virtual machine called Ethereum Virtual Machine (EVM) that provides runtime environment for smart contracts based on Ethereum. The Ethereum Virtual Machine is the fundamental consensus mechanism for ethereum. It is sandboxed and completely isolated from the file system, main network, or the other processes of the host computer system and, it is a perfect testing environment. Every node on the ethereum network runs their own EVM and implementation and is capable of executing the same instructions. Each ethereum node in the network has a ethereum addresses that are composed of the prefix "0x", a common identifier for hexadecimal, concatenated with the rightmost 20 bytes of the Keccak-256 hash. The Ethereum blockchain applications are referred to as DApps [10] (decentralized application) as they are based on the decentralized EVMs and its associated smart contracts.

4. PROPOSED SYSTEM

In this section our proposed system is explained with system architecture, along with various modules and overall system data flow.

4.1 SYSTEM ARCHITECTURE

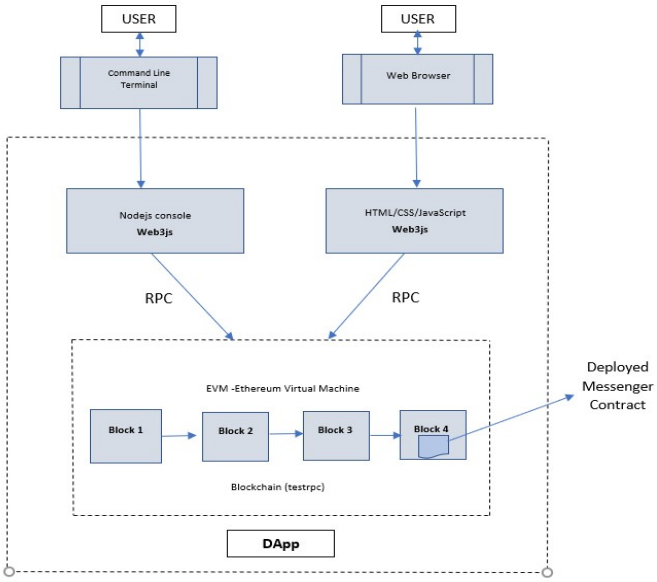

Figure 1: System Architecture

The system architecture comprises of a NodeJS server for implementing the IPFS upload functionality. It has a ethereum blockchain API which is made to run on localhost with a set of defined ethereum accounts. The data to/from the IPFS storage is added, retrieved through the web front end by means of specifying the file's hash. For web front end we have used HTML and JS. For back end operations we use NodeJS, Java.

The Modules for the proposed system are explained in the following sections:

4.2. COMPONENTS FOR EMBEDDING HASH INTO THE DATA

The data to be deployed must be embedded with a hash value to identify it uniquely. The hash value embedding in to the data is done via Least Significant Bit [11] (LSB) encode technique. Basically, data are arranged in bytes. Each byte is composed of eight bytes. Least Significant Bit encode is a type of steganography which involves modifying the least-significant bit within each byte of the data. By means of changing the least-significant bit (LSB) throughout

the entire data does not change the overall value very much. A Java code is used to embedded the hash value in to the song and to separate the hash from a song. This can be done by means of LSBEncode and LSBDecode methods respectively.

4.3.LEAST Significant BIT TECHNIQUE

Least Significant Bit substitution is the process of adjusting the least significant bit of a data. It is a simple approach which embeds a message into a data. The LSB varies according to the number of bits in a data. For instance, 90_{10} for which the binary form is 1011010_2. By changing the LSB from 0 to 1 will not make very much change in that value as 1011011_2 which is 91_{10} . Thus, over an array of data this change of multiple single bits is negligible and unnoticed. In similar manner a secret hash value (256 bits) will be converted into binary and then embedded with the song data. The embedded secret value is unaltered and unknown to the persons who are downloading, having it. The data is then uploaded to the IPFS and then shared through the Ethereum Blockchain.

4.3.1 LSB DECODE AND ENCODE

The Least Significant Bit Encode and Decode functionality can be easily done by a Java Code. Encode takes the Song and text file consisting of the secret message to be embedded into the song as inputs and produces a new song file embedded with the hash identity. Decode takes original song and embedded song as inputs and produces the hash code (separated from the file) as output.

4.3.2 THE PRETREATMENT OF THE SONG

The data owner must need to have the song file in such a format (requirements) that - 1: the audio file in a wav format, 2: audio with mono channel 3: audio volume frequency in the range of 44.1 kHz. The Song file will be then encoded with the text file contents (hash code) which is used for identifying the person uniquely by LSBEncode.

5. DEPLOYING DATA TO IPFS STORAGE

InterPlanetary File System (IPFS) [12] is a protocol and eponymous network designed to create a content addressable, peer-to-peer method of storing and sharing hypermedia in a distributed file system. Our proposed system uses IPFS for decentralized storage of files even with large size. Each of the file deployed in IPFS is identified using its hash value. IPFS uses peer-to-peer model of decentralized network for sharing hypermedia (non-linear medium of information includes graphics, audio, video, plaintexts, etc.). IPFS has advantage of no single point of failure, and nodes do not need to trust each other, except for every node they are connected to. IPFS instead of referring to objects (such as pictures, articles, videos) by which server they are stored on, it refers to everything by the hash on that particular. IPFS stores each data as objects the structure of a Merkle DAG [13] which signifies that the data is cryptographically authenticated data structure that uses cryptographic hashes to address content.

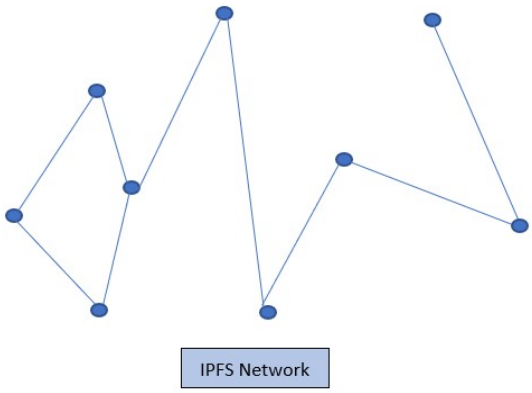

Figure 2: An IPFS Network

Instead to creating hash for the file that is being uploaded, rather IPFS creates hash for the contents of the file. This avoids creating hash for same files when they are uploaded more than once. IPFS represents the hash of files and objects using Multihash [14] format and Base58 [15] encoding. The hash corresponds with the Secure Hash Algorithm and length 46 bytes. When the files have been uploaded into the IPFS network, the IPFS protocol returns a hash for each of the file being uploaded. The hash code cryptographically represents the file, which is (all files being uploaded) seen as an object in the IPFS network. IPFS uses a gateway through which the network objects can be downloaded by entering the hash for that object (file).

The IPFS module represents initialization of the IPFS node and setup of the Peer ID which is a 46-bit hash identity that is used to identify that particular IPFS node in the decentralized IPFS network. In our proposed system we use IPFS version 0.4.13. A NodeJS and a frontend web page can be used to upload files to IPFS network by simply choosing the

song file which is already embedded with the hash value given by section 4.

6. ETHEREUM BLOCKCHAIN USAGE

The use of Ethereum blockchain is structured into back-end, front-end. At the back-end we have an active localhost server which runs on testrpc command that has a set of active ethereum accounts on the host computer. The testrpc command when executed displays a list of active ethereum accounts and their associated private keys in the console in the host computer. The testrpc command also performs the mining operation automatically by observing the changes in the smart contract that are compiled and executed over a solidity [16] development environment. The testrpc command observes changes and automatically redeploys the smart contracts without being controlled or initiated by someone. At the front-end we have a web page that is web3 [17] interface injected into the smart contract and responses to the events that happening around the web page.

The web page has the script that runs for he ethereum. In the web page we import the smart contract Abstract Binary Interface [18] (ABI) and the contract deployed address so that it is used to call the functions in the smart contract. The data owner will be assigned with ethereum account 0 as a default account. The web page is used to send a message from our default ethereum account to another ethereum account. At the front end we have also Metamask [19] which is the browser extension that displays the active ethereum accounts and the transaction logs along with their ether balances. For every account in the ethereum network, we will be generating unique hash code for identifying that account. The data owner initially embedded this hash value into the song and then deploys the file in to the IPFS system. By doing so we will obtain a hash for that particular file that has been deployed into the IPFS. The data owner then shares the hash with the ethereum account. The data owner then shares the file hash to that account by means of sending a message to that account. This transaction will be logged. The hash value sent will be used as a URL to download the uploaded file. The account that uses the URL to download the file will have a copy of song that is uniquely embedded for that account/person. If the person who downloaded that file illegally pirates that file, the pirated version will be having the hash identity (initially obtained from module 4) embedded in it. The data owner when knows this illegal transfer, download that particular pirated version and then decodes the song to get the hash value.

The data owner then compares that hash value with the ethereum accounts hash value he/she already have and identifies the account/person who pirated the song file illegally. The data owner after finding the actual account/person who pirated the file illegally avoids to share any other further data with him/her thereby reducing the chances of piracy. LSBDecode is used for decoding the hash from the Song file that is pirated illegally.

6.1 TECHNICAL DETAILS

- **Ethereum Blockchain Network:** It is used to verify the ethereum accounts in back-end. The active ethereum accounts runs on the top of the blockchain. The blockchain logs all the transactions taken place. Smart contract is being deployed and run in the Ethereum Virtual Machine (EVM) enabling auto changes and corrections.
- **Smart Contract:** They are compiled and deployed once. The contracts are re-deployed automatically whenever any changes occur. It is used for ensuring the messaging functionality to send the message to the particular ethereum account.
- **IPFS web server:** The IPFS web server is being running at background and provides the API for uploading files into the InterPlanetary File System. The web page has a NodeJS server that locally hosts a IPFS web page that has file upload functionality and returns a hash file for that file uploaded.
- **SHA-256:** The Secure Hash Algorithm is used for the purpose of identifying, embedding the hash in to the file, which is uploaded to the IPFS and then shared to a particular user for the ethereum nodes in the network.
- **LSB Encode and Decode:** The hash identity is being embedded into the song, and then separated from the song by means of the LSB Encode and LSB Decode techniques.

7. DATAFLOW

The data flow is shown in the Fig: 3, as the Data owner initially embedded the secret hash value which

is (32 bytes) and by means of Secure Hash Algorithm and then deploys that data over the InterPlanetary File System.

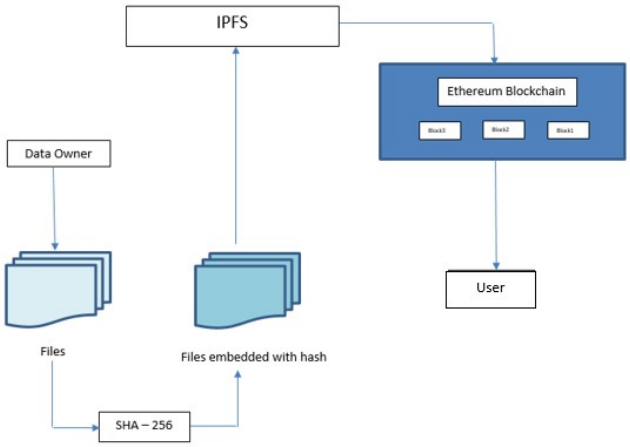

Figure: 3 Data flow from Data owner to User

The IPFS creates hash for each data being uploaded. That hash value will be shared among the user through the Ethereum Blockchain network. Each of the transaction will be logged by the data owner. The Data owner ensures data being shared legally by means of having unique hashes for each of the data being sent to other nodes in the Ethereum Blockchain. If the data is illegally pirated through any of the nodes, the data owner will have the pirated version of the data and then separates the hash code from that data to get the hash and then compares it with the hash log he/she has. Then, the ethereum account which pirated the data will be found and data sharing for that node will be not permitted thereafter.

8. CONCLUSION

We have proposed a system that can be used to avoid data piracy in artwork distribution with Blockchain technology. Blockchain gives us truly distributes peer-to-peer systems and ability to interact with peers in a trustless environment and smart contract allows us to automate complex multi-step process. By incorporating these technologies, we can have a better artwork distribution system that tracks and avoids data piracy.

Further, using blockchain we can ensure that all the information sent is cryptographically secured and encrypted. The proposed approach empowers the combination of blockchain technology in artwork distribution to have a feasible and secure solution for data sharing to be made legally.

REFERENCES

1. J. Kelly and A. Williams, (2016). Forty Big Banks Test – Blockchain-Based Bond Trading System. http://www.nytimes.com/reuters/2016/03/02/business/02reuters-banking-blockchain-bonds.html
2. S.Lacey (2016). The Energy Blockchain: How Bitcoin could be a Catalyst for the Distributed Grid.http://www.greentechmedia.com/articles/read/the-energy-blockchain-could-bitcoin-be-a-catalyst-for-the-distributed-grid
3. Bayu Adhi Tama, Bruno Joachim Kweka. A Critical Review of Blockchain and its Application. ICECOS, 2017. DOI: 978-1-4799-7675-1/17
4. Sonali Patel, Tejal Chavan, P.R.Deshmukh. Reduced Share size Audio Secret Sharing, ICPC, 2015, DOI: -1-4799-6272-3/15
5. Thomas Bocek, Bruno B. Rodrigues. Blockchains Everywhere - A use case of Blockchains in the Pharma Supply Chain, IFIP, 2017, DOI: 978-3-901882-89-0.
6. Yongle Chen, Hui Li, Kejiao Li and Jiyang Zhang, An improved P2P File System scheme based on IPFS and Blockchain, IEEE, International Conference on Big Data, DOI: 978-1-5386-2715-0/17
7. Jiawen Kang, Rong Yue, Xumin Huang, Sabita Maharajan, Enabling Localized Peer-to-Peer Electricity trading among Plug-in Hybrid Electric Vehicles Using Consortium Blockchain, IEEE, 2017, DOI: 10.1109/2709784.
8. Xueping Liang, Juan Zhao, Sachin Shetty, Jihong Liu, Danyi Li1, Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications, 2017, IEEE, DOI: 978-1-5386-3531-5/17.
9. Ethereum, Wikipedia. [Online] available at: https://en.wikipedia.org/wiki/Ethereum
10. Ethereum Dapp stackexchange, [online] available at: https://ethereum.stackexchange.com/questions/383/what-is-a-dapp
11. LSB Steganography, [online] available at: https://cyfor.engineering.nyu.edu/topic/02-lsb-steganography/
12. InterPlanetary File System (IPFS), [online] available at: https://ipfs.io, https://medium.com/@ConsenSys/an-introduction-to-ipfs-9bba4860abd0
13. Merkle DAG, [online] available at: https://github.com/noffle/ipfs-merkle-dag-node

14. Multihash, [online] available at: https://github.com/noffle/ipfs-merkle-dag-node
15. Base58 encoding, [online] available at: https://en.wikipedia.org/wiki/Base58
16. Solidity, [online] available at: https://github.com/ethereum/solidity
17. Remix Solidity IDE : https://remix.ethereum.org/
18. Ethereum Web3, [online] available https://github.com/ethereum/web3.js
19. Abstract Binary Interface, [online] available at: https://en.wikipedia.org/wiki/Application_binary_interface
20. Metamask, [online] available at: https://github.com/MetaMask