Blockchain based mechanism to eliminate frauds and tampering of land records

Devi D[1]*, *Sai Rohith* G, *Shri Hari* S, and *Sri Ramachandar* K
[1]Department of Electronics and Communication Engineering, Sri Krishna College of Engineering and Technology, India

Abstract: Data tampering and fraud in land records have increased drastically in the modern world. A data storage model using Blockchain and Interplanetary File System (IPFS) is proposed in this work. Land records and the farmer's information are stored inside the Interplanetary file system. To avoid data faking, the hash address of the respective data generated by IPFS is stored in the blockchain. This proposed system when deployed on a large scale can outperform the existing methods of securing user data. One of the latest technological advancements in the software industry is the innovation of Blockchain Technology. This new technology has opened up a new business relationship platform that delivers feasibility, protection, and cheap rates. It provides a new foundation of trust for transactions that can facilitate a very streamlined workflow and a faster economy.

1 Introduction

With the world around us moving at a faster pace, the security of our possessions has become a major concern. Illegal acquisition of agricultural land and misuse of land records had been increased. However, these problems can be solved with the power of new technologies like blockchain. A blockchain is a decentralized, distributed, and made up of several records called blocks that are connected through encryption. It is a peer-to-peer network that is handled anonymously by a mass collaboration of people (nodes) with self-interests. Blockchain manages a limited number of blockchain operating systems, including the blockchain cryptographic hash that preceded them.

A blockchain-like protocol was first proposed by Cryptographer David Chaum in 1982. Merkle trees were incorporated in blockchain by Haber, Stornetta, and Dave Bayer in 1992 which drastically improved efficiency. However, blockchain was first conceptualized and devised as bitcoin, by Satoshi Nakamoto in 2009 [1]. This has been a breakthrough in electronic cash systems and cryptocurrencies. Blockchain was primarily created to monitor digital currency transactions, however, its robustness and immutable nature have proved useful in many domains. Since it is an append-only ledger, and as the number of blocks generated increases, it becomes very slow and highly ineffective for storing real-time data [2]. So blockchain technology is not applied directly. In this paper, blockchain is combined with an Interplanetary file system (IPFS) to gain security and reliability. The contents of this work are as shown, the consecutive section provides the related work and Section 3 gives an overview of the architecture of our proposed model. Section 4 explains the working analysis of our model in detail, the work outcomes, and the conclusion is given in the final section.

2 Related Work

An overview of related work that leverages access control methods and blockchain-based data storage is proposed in this section.

2.1 Data storage solutions:

Blockchain provides us with an unchangeable database that allows us to collect transactions and never delete or modify them. This enhances security, however, the real-time costs of storing a massive amount of data inside the blockchain is not feasible [3]. For instance, according to etherscan an average size of a block in July 2020 is 40kb.1kb costs around 0.032 ETH or 828 INR when using Ethereum. Hence the developer should decide which data to keep off-chain and which data to keep on-chain.

*Corresponding author: devi@skcet.ac.in

2.2 Data access approach:

Traditional access control mechanisms rely on a centralized database system to store and access data. However, under a centralized system, an organization maintains the identities and access rights of the user. Some large media file-sharing systems, such as BitTorrent, KaZaA, and Napster, are introduced to store massive data. Alam et al presented the Interplanetary Wayback as a permanent Web archive to distribute data files into the IPFS network [4]. Header and payload are split for every response record, then they are disseminated into IPFS. The average indexing rate can be boosted by this method. However, various studies show that blockchain can be used as an access control manager for distributed systems. We have proposed an idea, where the data collected in the decentralized database in which a third-party application has control over it. This is achieved by combining blockchain and IPFS with an off-chain solution to track user identity and corresponding document identity which is described by a tuple.

The key advantage of using blockchain is that the user identity data cannot tamper inside this third-party application. The smart contract manages and provides access and control [5].

2.3. Privacy:

The key concept behind privacy in the blockchain is public and private key cryptography. These systems use asymmetric cryptography to secure transactions [6]. Private keys are used to protect the user data through digital signature, hence adding a layer of authentication. Ethereum uses the Elliptic Curve Digital Signature Algorithm (a version of the Digital Signature Algorithm) which uses elliptic curve cryptography, to generate key-value pairs. The proof of work function used in Ethereum is Ethash (256-bit), belonging to the keccak family (same family as SHA-3) [7].

3 Architecture Overview

This section provides an introduction to the data storage model and how smart contracts control the flow of data and add a layer of security.

3.1 System Overview:

A highly secure and reliable system has been designed to store farmer details and their land details using blockchain as the layer for monitoring transactions, Interplanetary File System for data storage, and a local database management system for tracking public user details and transaction hash codes as shown in Figure 3.1. Currently, all the records are manually maintained by the department leading to difficulty in access and retrieve the records, also increasing vulnerability. The user details and land information are obtained through the government servers or any offline methods devised by the government. The data is processed and stored by the data model architecture.

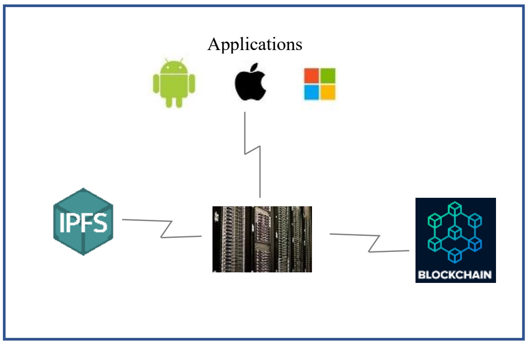


Fig. 3.1 System Overview

3.2 Interplanetary File System (IPFS):

The state-of-the-art file storage system which is decentralized in terms of architecture and a robust system called The Interplanetary File System (IPFS) is being implemented. IPFS provides a block storage model that is content-addressed hyperlinks and has high throughput [8]. Hence a generalized Merkle directed acyclic graph (Merkle DAG) is formed, a data architecture onto versioned blockchains, file systems and permanent web are built. IPFS combines an induced exchange of blocks, a distributed hash table, and a self-certifying name box. There is no single point of failure in IPFS since there is no need for two nodes to trust each other.

IPFS envisions to develop a distributed web, making it immune to central server attacks. In IPFS the system relies on content-based addressing rather than location-based addressing followed in HTTP protocol. IPFS uses a Distributed Hash Table or DHT to store data. This system leverages physical proximity. Upon receipt, a network of friends with content belonging to hash is called and the content is obtained straight via the nodes holding the necessary information instead of connecting to a standalone server. Data transfer between nodes in a network is as same as to the mechanism in a BitTorrent.

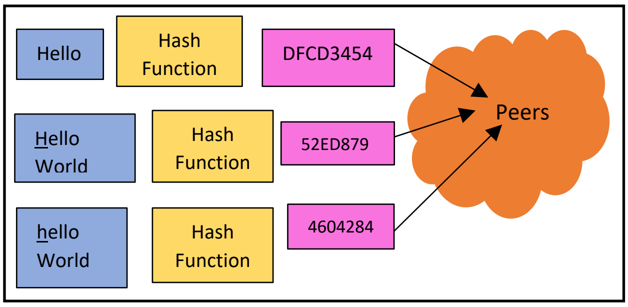


Fig. 3.2 Working of hashing in IPFS

Since IPFS uses hashing to store data as shown in Figure 3.2, data duplication is avoided, hence increasing efficiency. Files greater than 256 kilo-bytes are split into

IPFS objects and their respective hash codes are generated and further linked into another IPFS object.

3.3 Ethereum and Smart Contract:

Ethereum is one of the most popular blockchain operations. Its custom currencies called tokens and native cryptocurrency ether are used to make transactions [9]. Its use of smart contracts and extensive integration make it highly secure and reliable. It can be used to create an alternative protocol to develop decentralized applications, with particular emphasis on scenarios where security for rarely used and small-scale applications, fast development time, and the ability of applications to interact very efficiently, are crucial. Ethereum has an intellectual basic layer: a built-in theme-based programming language, allowing developers to write contracts and decentralized apps where they can define their own configuration rules for state transition tasks states, forms of trade and property. The Ethereum protocols follow the principles, i.e., modularity, agility, simplicity, universality, non-discrimination, and non-censorship. No entity has control over the execution in a decentralized network. Firstly, since the business logic present in the contract is agreed upon by consensus the resultant from the execution of code can be trusted. Secondly, due to proof of work, consensus, and append-only ledger in Ethereum protocol, code tampering or changing is not possible.

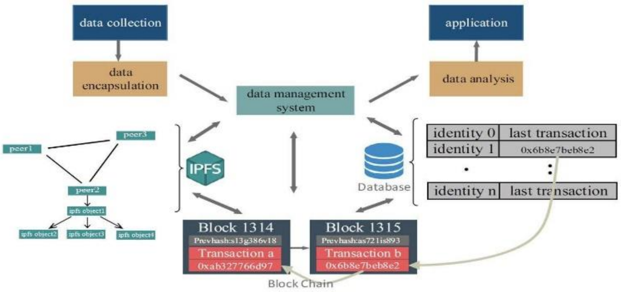
Solidity with compiler version 0.4.22 is used for writing and compiling the contract and storing the access control list. Solidity is typed statically and has various variables, functions, and structures. It avails msg.sender which is implicitly available and holds the path of the user who is initiating the transaction, thus ensuring authenticity [10]. The architecture named OwnerDetails is depicted in the contract containing two variables of type address, called PO_hash and data_hash, a process to map called _approve, and a variable array of paths named user. Variable sender is mapped to an example of the structure OwnerDetails, called map.

Only some functions are accessible by the user. The validation is done using if- statements. If the resultant of the if-statement is false then the contract does not update the transaction, hence the state of the contract is not altered. Those functions which do not make any changes to the smart contract are called constant functions, and they do not require any costs to execute the transaction (since they do not alter the state of the contract).

addIpfsHash: Cryptographic hashes help us identify ipfs files. A hash value of bytes32 type is accepted as an only argument in this function. It checks whether the hash represents some data that already exists in the network or whether the file is empty. If either of these checks fails, the transaction gets declined. Otherwise, the hash is stored in the OwnerDetails structure and as a key in map mapping, and it's the key holds for the owner which routes to msg. sender. The contract now commits this obtained data to the database.

retriveFileAccess: This function accepts a PO_hash and a data_hash representing an IPFS hash as an argument. The transaction fails if the entered hash doesn't match the mappings. Otherwise, the relevant data is retrieved utilizing the hash variables as the key from the storage. The msg.sender maps the recorded data of ownerDetails at an instance and approached to the destined path via a map.

In our proposed system a feature has been added in the smart contract, in which the hash address of the previous owner along with the data from whom the land was bought or inherited is stored as shown in Figure 3.3. This allows us to track the hierarchy and retrieve several useful information about the land.



identity 0	last transaction
identity 1	0x6b8e7beb8e2
.	:
identity n	last transaction

Fig. 3.3 A proposed model overview

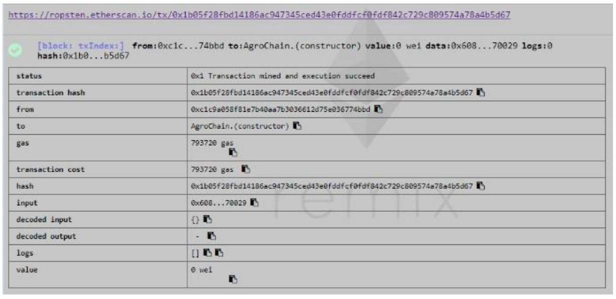
3.4 Data Model Architecture:

A highly secure system is designed to store farmer details and their land details. Currently, all the records are manually maintained by the department leading to difficulty in accessing and retrieving the records, also increasing vulnerability. Firstly, the user details and land information are obtained through the government offices, encapsulate the data, process the above data, and store them into IPFS as shown in Figure 3.3. The user can verify that the correct information is uploaded by encrypting it with his private key and various offline methods, which can be devised by the government. The authenticity of the data obtained in the system is ensured by storing the hash generated by the IPFS in the blockchain. The transaction hash generated by the blockchain after completing a successful transaction along with the public details uniquely identifies a person, which is then stored in a database management system to track the hash codes generated during this process.

When extracting user information, the system uses the transaction hash to query transactions and obtain the IPFS hash address. With this hash, the data from IPFS can be retrieved. The data analysis module analyses the data taken from the database management system and returns it to the application layer.

4 Working Principle

This model uses DAG in IPFS to manage the user data. The PropertyOwner object stores the related details of the farmer and his owned property [11]**.** Firstly, the smart contract is deployed in ropsten test network and injected web3 environment. In our testing, the contract was deployed in the block 8950103.



```
https://ropsten.etherscan.io/tx/0x1b05f28fbd14186ac947345ced43e0fddfcf0fdf842c729c809574a78a4b5d67

[block: txIndex:] from:0xc1c...74bbd to:AgroChain.(constructor) value:0 wei data:0x608...70029 logs:0
hash:0x1b0...b5d67

status               0x1 Transaction mined and execution succeed
transaction hash     0x1b05f28fbd14186ac947345ced43e0fddfcf0fdf842c729c809574a78a4b5d67
from                 0xc1c9a058f81e7b40aa7b3036612d75e036774bbd
to                   AgroChain.(constructor)
gas                  793720 gas
transaction cost     793720 gas
hash                 0x1b05f28fbd14186ac947345ced43e0fddfcf0fdf842c729c809574a78a4b5d67
input                0x608...70029
decoded input        {}
decoded output        -
logs                 []
value                0 wei
```

Fig. 4.1 Transaction log after deploying a smart contract

The data attributes are used to store objects and related information. Links property is used to store the previous owner of the particular record along with the current record hash address of the data in IPFS

```
   {
   "data": {"type": "PropertyOwner", "user_id ":
   "0x567b"},
   "links": [
   {"hash":"QmQZzTMN2X54SxC2jMuAug6Qcz1KYS
   5ZB12i3gGusvn", "name": "record_11010 ",
   "size":"7987"},
          {"hash":"QmUgeQaCjhZ8V42DKBLt
          TCvzxojUt-LpaX6QmXX9rTLNnxF", "name":
          "record_0850", "size":"5765"}
          ]
}
```

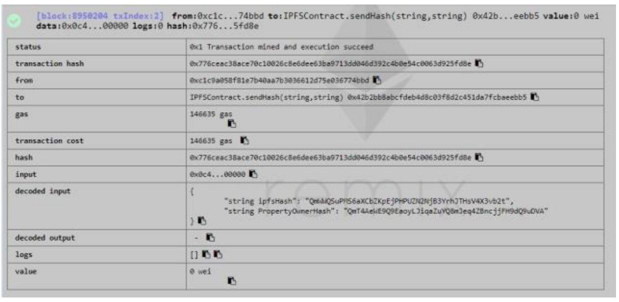
The data obtained during data collections after processing is then stored in the IPFS as shown in Figure 4.1.

```
Algorithm: Writing data to IPFS
Input: data, hash of the previous owner. Output: data
IPFS hash, PO IPFS hash
1: get old PO_data → get old property owner data with
old PropertyOwner hash (old PO_hash);
2: data_hash → store the data into IPFS and
obtain the data hash;
3: generate new PO_hash → link the old PO_hash with
new PO_data and generate a new PO_hash;
4: return data_hash, PO_hash;
```

There is a risk of storing the PropertyOwner IPFS hash address directly into the blockchain. The tamper only needs to modify the latest blockchain transaction hash and links the contents of the transaction to the spoofed data IPFS hash address to achieve the purpose of tampering with the data. To deal with this problem, the hash of the newly stored data is uploaded beside the IPFS hash of PropertyOwner. The hash address of the provenance data can be matched with the PropertyOwner while querying. Each time, when the data is collected in IPFS, the updated PropertyOwner hash address and the hash path of the corresponding data block are reported to the management system. The management system encapsulates these addresses into a transaction and write it into the blockchain. The transaction is as follows: Encapsulated as blockchain transaction data:

```
{
 "jsonrpc": "2.0", "method": eth_sendTransaction", "id":
1, "params": [{
            "from":"0xb60e8dd655152be8058bb8e
            b970870f07233155",
            "to":"0xd456odd67c5d32be8058bb8eb
            970870f07244567", "gas": "0x76c0",
            "gasPrice": "0x9184e-72a000", "value":
            "0x9184e72a",
            "data":"0x3836303731393032333939 3
                      53831383b33323b3235313b31
                      30323435313b302e32353b313
                      438323739393633353b333935
                      372e35373231343b4e3b313136
                      32302e39393030353b45"
            }]

   }
```

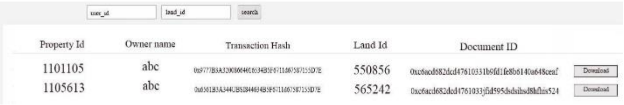


```
[block:8950204 txIndex:2] from:0xc1c...74bbd to:IPFSContract.sendHash(string,string) 0x42b...eebb5 value:0 wei
data:0x0c4...00000 logs:0 hash:0x776...5fd8e

status               0x1 Transaction mined and execution succeed
transaction hash     0x776ceac38ace70c10026c8e6dee63ba9713dd046d392c4b0e54c0063d925fd8e
from                 0xc1c9a058f81e7b40aa7b3036612d75e036774bbd
to                   IPFSContract.sendHash(string,string) 0x42b2bb8abcfdeb4d8c03f8d2c451da7fcbaeebb5
gas                  146635 gas
transaction cost     146635 gas
hash                 0x776ceac38ace70c10026c8e6dee63ba9713dd046d392c4b0e54c0063d925fd8e
input                0x0c4...00000
decoded input        {
                             "string ipfsHash": "QmWWQSuPMS6aXCbZKpEjPHPUZN2NjB3YrhJTHsV4X3vb2t",
                             "string PropertyOwnerHash": "QmT4AeWE9Q9EaoyLJiqaZuYQ8mJeq4ZBncjjFH9dQ9uDVA"
                     }
decoded output        -
logs                 []
value                0 wei
```

Fig. 4.2 Transaction log after storing the IPFS and PO_hash

The transaction hash is queried from the storage and the IPFS data hash is accessed via the blockchain, which can be utilized to retrieve the original data as demonstrated in Figure 4.2 and Figure 4.3.



Property Id	Owner name	Transaction Hash	Land Id	Document ID	
1101105	abc	0x9777B3A3200B6640165 34B5F6711467587155D7E	550856	0xc6acd682dcd47610331b9fd1fe8b6140a648ceaf	Download
1105613	abc	0x6561B3A344UBS8446634B5F6711467587155D7E	565242	0xc6acd682dcd4761033jfid595dsdsihsd8hfhis524	Download

Figure 4.3 Data analysis after querying owner information from database.

```
Algorithm: Data query algorithm Input: user_id, public
details (other parameters)
Output: result (success or fail)
1: Tx_hash → Query transaction hash from the database
by user_Id and other parameters;
2: old PO_hash → Get old PropertyOwner IPFS
hash from blockchain by Tx_hash;
3: if verification (PropertyOwner hash) == success
        Return data, success;
   Else
        Return fail;
```

https://doi.org/10.1051/itmconf/20213701011

5 Experimental Analysis

Ethereum v1.9 is used as blockchain and IPFS v0.5.0 to store data. The database for tracking the user information is Oracle v18.4.0.0.0. This model was deployed in 6 machines, with each machine having a minimum configuration of 1.8GHz Intel Processor with 8GB ram and windows OS.

5.1 Case Study

The comparative study of IPFS storage and model based on Blockchain is being researched in detail determining their efficiency on real-time scenarios. The time taken to process the uploaded data is shown in horizontal axis also determining the amount of data i.e., the records that are uploaded per minute in kilobytes is represented in vertical axis.

The proposed model is designed to store records efficiently and secure transactions using blockchain. The system was tested by uploading data every two minutes**.** The processing efficiency of our system increases exponentially as the size of data uploaded increases.

The experimental observations are plotted in Figure 5.1. We expect that IPFS consumes less time however in real-time scenarios when a single person uploads a small file, we observe that both the speeds are more or less similar in both Blockchain and IPFS but as of enterprise-level, IPFS storage must be preferred where the number of transactions is exponentially increased. The upload transaction is depicted in Figure 5.2.

During the download speed testing, we observed deflections in both IPFS and Blockchain, and in contrast, IPFS performance is significantly higher. As the size of the files scales up, we note that the download speed of Blockchain becomes more stable while IPFS even facing deflections serves being a better performant as shown in Figure 5.3. The comparison of upload and download speed between IPFS and Blockchain is depicted in Table 5.1 to show the variations as the file size scales up.

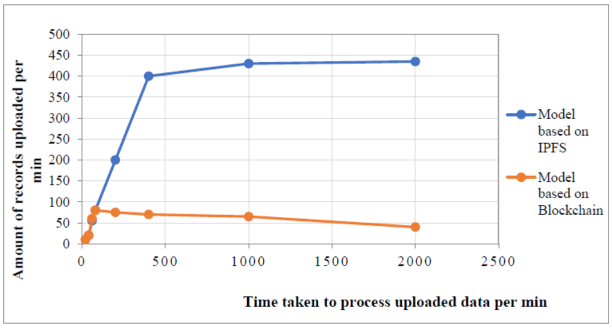


Fig. 5.1. Efficiency of IPFS in comparison with Blockchain

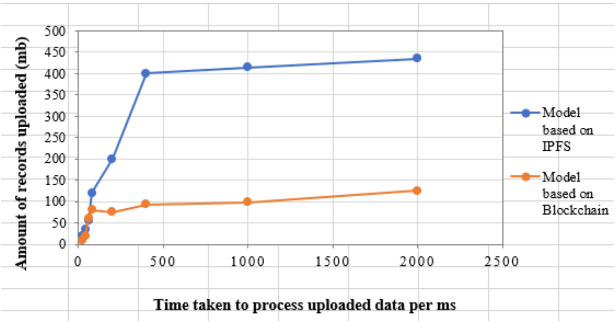


Fig. 5.2. Comparison of upload transactions between IPFS and Blockchain

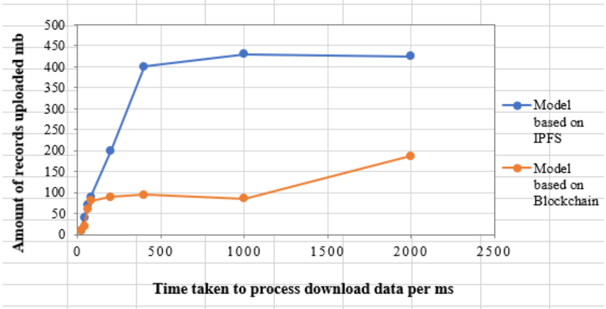


Fig. 5.3. Comparison of download transactions between IPFS and Blockchain

Blockchain (ms)	IPFS (ms)
Upload	
10	20
20	35
60	55
80	120
75	200
93	400
98	415
125	435
Download	
10	10
20	40
60	70
80	90
90	200
95	400
85	430
187	425

Table 5.1. Comparison of upload and download speed between IPFS and Blockchain

6 Results

We have designed an application to maintain the records and protect against frauds and data tampering. After manipulating the data query algorithm, the user could access secure data in the blockchain [12]. This can also help in monitoring the hierarchy of the property by keeping track of the previous transactions in the chain.

Our proposed system with IPFS and Blockchain has shown improved speeds compared to standalone Blockchain use cases. This system when deployed on large scale, would help to attain good real-time speed and reliable security via Blockchain. As we intend to use a centralized database to monitor the transactions, it becomes highly feasible to query any transaction done in past. Hence, it provides a reliable approach to deliver the power of a centralized database alongside the certainty of IPFS and Blockchain.

7 Conclusions

This paper has addressed the requirement to store the data of the user without the risk of data tampering using blockchain technology preventing forging of records and overcoming the hacking and manipulation of centralized databases. As discussed above, storing files directly in blockchain is not feasible. For this purpose, the hash address of the data is stored in the blockchain, along with a third-party database management system to monitor and query the transactions. The delay observed during this process is primarily due to the engagement with blockchain, which may hinder scalability and convenience in real-time. This mode of storing the records in Blockchain with the help IPFS makes a reliable and trustworthy system rather than the existing centralized mode of storing these records. The significant access and transparency are the key to correspond the proposed model.

8 References

1. S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/en/bitcoin-paper (2008)
2. Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,* IEEE, 6th International Congress on Big Data, 557-564 (2017)
3. M. Steichen, B. Fiz, R. Norvill, R. State, and W. Shbair, *Blockchain-Based, Decentralized Access Control for IPFS*, Conference on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, IEEE, 1499-1506 (2018)
4. S. Alam, M. Kelly, and M.L. Nelson, *InterPlanetary Wayback: The Permanent Web Archive*, Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL, 273-274 June (2016)
5. SmartContracts, https://docs.soliditylang.org/en/v0.4.24
6. Privacy and Blockchain: https://en.wikipedia.org/wiki/Privacy_and_blockchain
7. J. Benet, IPFS - Content Addressed, Versioned, P2P File System (DRAFT3), https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf, (2014)
8. H. Gilbert and H. Handschuh, *Security Analysis of SHA-256 and Sisters*, International Workshop on Selected Areas in Cryptography, Springer, **3006**, 175-193 (2003)
9. Ethereum Documentation, https://ethereum.org/en/whitepaper/
10. Solidity 0.7.4 documentation, https://docs.soliditylang.org/en/v0.7.4/
11. J.T. Hao, Y. Sun and H. Luo, *A Safe and Efficient Storage Scheme Based on Blockchain and IPFS for Agricultural Products Tracking*, Journal of Computers, **29**, 158-167 (2018)
12. Transaction made in ropsten test network, https://ropsten.etherscan.io/tx/0x1b05f28fbd14186ac947345ced43e0fddfcf0fdf842c729c809574a78a4b5d67