International Journal of Pure and Applied Mathematics
Volume 119 No. 15 2018, 1437-1442
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

A Blockchain and IPFS based framework for secure Research record keeping

Rajalakshmi A[1], Lakshmy K V[2], Sindhu M[3]and Amritha P P[4]
[1,2,3,4]TIFAC-CORE in Cyber Security,
Amrita School of Engineering, Coimbatore,
Amrita VishwaVidyapeetham,
India.
{[1]*lakshminan13@gmail.com*, [2]*kv_lakshmy@cb.amrita.edu*, [3]*m_sindhu@cb.amrita.edu*,
[4]*pp_amritha@cb.amrita.edu*}

***Abstract*—Research record keeping in academics is important in order to ensure proper planning, management and execution of research work. With the rapid development of technology and increasing amount of information records, there are huge chances for information leakage and record tampering, which is a serious threat to privacy and authenticity of the research records. This information when stored in a central server may lead to problems in efficiency. So there is a need for a distributed system, which is both efficient and secure. Blockchain is the emerging technology which attempts to solve these issues by creating tamper proof event of records in a distributed environment. IPFS is a protocol designed to store hypermedia in a peer-to-peer distributed file storage with content-addressability. The framework proposed in this paper attempts to combine both these technologies and other traditional encryption methods to create a secure, tamper proof model of academic research record keeping with access control methods. Furthermore, the system utilizes ethereum smart contracts to store the provenance metadata information retrieved from the IPFS file system to the blockchain network, to create tamper-proof records for further auditing purposes.**

***Keywords*—Academic research record keeping, blockchain, IPFS, ethereum, smart contract, provenance metadata,tamperproof.**

I. Introduction

Academic research record keeping is important for the research planning and management, replication of results, documentation of collaborations, publishing and peer review, and for complying with governmental and institutional rules and regulations. Good research records consist of much more than just research data. They include protocol description, data manipulation and analysis procedures, personal and group interpretation of the results, and important communications and group decisions among collaborators. So the data must be confidential, secure and tamper proof in order to avoid any discrepancies. While considering academic research, the Principal Investigator(PI) is the main actor who ensures proper research planning, management and execution of the ongoing research. Documents like proposal of funding agencies, project reports, memorandum of understanding, minutes of meeting etc., must be preserved in a secure and tamper proof environment without leakage of information, since they are very critical. A slight variation or modification in these documents may lead to serious consequences.

Traditional databases can be used for storage of this data. Since traditional database management systems involve a central authority to take control of a large amount of data, one cannot trust on the confidentiality, integrity and authenticity of the data. So, there are many issues in centralized systems such as Denial of Service(DOS) attacks and single point of failure. Here comes a need of a distributed technology which ensures the authenticity, confidentiality and integrity of data.

Blockchain technology can create tamper-proof, secure record of events in a distributed, peer-to-peer network of several nodes of computers. The cryptocurrency based transaction system like bitcoin is based on this technology. Blockchain ensures anonymity and security of the users involved in the transactions. Blockchain consists of a growing list of blocks which are linked by cryptographic algorithms. It is based on the Distributed Ledger Technology (DLT) which is a system for recording digital transactions in a distributed storage with no centralized data stores.

The distributed ledger technology can be used to write smart contracts or digital contracts or blockchain contracts which are self-executing contracts that can be converted to computer code with the help of certain platforms, and can be replicated, shared and supervised by network of computers that run on the blockchain. Smart contracts avoids middleman by automatically defining and enforcing rules and obligations made by the parties in the ledger. While blockchain can be used for storage of less amount of data like transaction metadata information, hash values etc., IPFS can be used as a peer-to-peer, distributed system to store hypermedia in large quantities.

InterPlanetary File System (IPFS) [3] is a peer-to-peer hypermedia protocol and distributed file system that is to replace the web of tomorrow. It has a block storage model with hyperlinks to address the contents forming a Merkle Directed Acyclic Graph (DAG). Since IPFS is distributed, it has no single point of failure.

There are many disadvantages of HTTP such as inefficiency, no historic versioning, and centralization. So IPFS overcomes the disadvantages of HTTP.

This paper presents a framework where the scientific research record keeping can be done in a secure, tamper proof environment using blockchain technology, IPFS and smart contracts. For the storage of documents such as project reports, memorandum of understanding, funding projects documents, attendance records, and minutes of meeting, IPFS is utilized,along with certain access control methods, since all participating nodes in the network need not necessarily be able to access all the important information. Methods like secret sharing [20] and asymmetric key cryptosystem can be implemented as additional functionality in the system for limiting the access structure only to certain users of the system. The provenance metadata information of the documents stored in IPFS is further uploaded to the blockchain in order to ensure the integrity of the information. The Principal Investigator(PI) can ensure that these documents are only accessed and modified only by intended users, who are allowed access, using this audit information on the blockchain.

The structure of the paper is as follows: The next section provides the necessary background information and the work which is related to the framework proposed in this paper. Section 3gives the overview of the framework proposed. Section 4 provides the analysis of the framework and Section 5 provides the conclusion and future work to be done.

II. Background and related work

Traditional centralized databases are mostly based on the client-server architecture, where the client can store entries in a central server, and can access updated copy of the information on each time of accessing the server. In contrast to this, blockchain is a growing list of blocks which are linked and secured using cryptographic algorithms. This technology was invented by Satoshi Nakamoto in 2008, for the purpose of using it in his cryptocurrency Bitcoin [1]. Each block in the blockchain contains list of transactions, hash of the previous block and hash of the current block. The first block in the blockchain is called the genesis block. Blockchain is a distributed ledger technology maintained by a peer-to-peer network consisting of nodes. For updating the distributed ledger, the participating nodes in the network should derive at a common consensus. The consensus protocol is the core and it decides how a blockchain works. Sankar LS et al., in [2] provides an analysis and study of various consensus protocols in blockchain and the feasibility and efficiency they provide in various platforms. Blockchain can be visualized as a trusted record keeping system based on archival science – an ancient science aimed for preservation of records [21].

IPFS [3] is the distributed and versioned file system which can connect many computing nodes with the same system of files and manage them by tracking their versions over time. IPFS has a special property of content addressing at the HTTP layer for the identification of files. IPFS represents a file by the hash on it, instead of representing it by which server it is stored on. The hash of files in IPFS always begins with "Qm" and the hash is actually a multihash. Name of files is IPFS is actually not a part of the IPFS object, so two files with different names and same content will have the same hash values. Ethereum blockchain's Merkle Patricia tree structure [5] can also be emulated as IPFS objects. For larger pieces of data to be stored on the ethereum blockchain, a larger amount of fee has to be paid, so only the hashes of files are stored on the ethereum blockchain rather than storing the whole file on it. Further, this hash of the file can be linked with the file on the IPFS to access it [4]. A novel zig-zag based storage model based on IPFS and blockchain is provided in [9] to address the issue of high-throughput for individual users in IPFS.

Smart contracts provide an easy way to access the ethereum blockchain. Ethereum smart contracts are written in a high-level coding language called Solidity [13] which is influenced by coding languages such as C++, javascript and Python. To develop ethereum smart contracts, Remix IDE [7] can be used, which is a browser based IDE. Another one is the Truffle framework [6], which supports built-in smart contract compilation, linking, deployment and binary management. It supports both public and private network deployment environments. The truffle framework has a one-click blockchain support mechanism called Ganache, which is an internal javascript implementation of the ethereum blockchain. It also has the support of front-end libraries with Drizzle. In order to run ethereum decentralized apps in the browser itself, without running a full node, MetaMask [8] can be used. The above tools can be combined for an effective ethereum decentralized application development.

Data provenance refers to the tracking and recording of the origins of data. It refers to the collection of history of data such as creation, attribution and data versioning. Provenance metadata is very important for forensics purposes and auditing. Blockchain can be used as a platform for provenance data management in a trustworthy manner. With the Open Provenance Model(OPM) [11] and ethereum smart contracts, immutable trials of data can be recorded [10]. ProvChain [12] is a distributed, cloud based data provenance architecture, which creates tamper-proof record of events by embedding the provenance records into the Blockchain as transactions. The system uses bitcoin Blockchain and Tierion API [14] is used to embed data records into Blockchain. Tierion API uses the Chainpoint standard [15], which is an open standard to create timestamp proof of any data record.

III. Proposed Framework

The users involved in the system are the Principal Investigator(PI) and Junior Research Fellow (JRF). The documents which are to be considered are project reports,

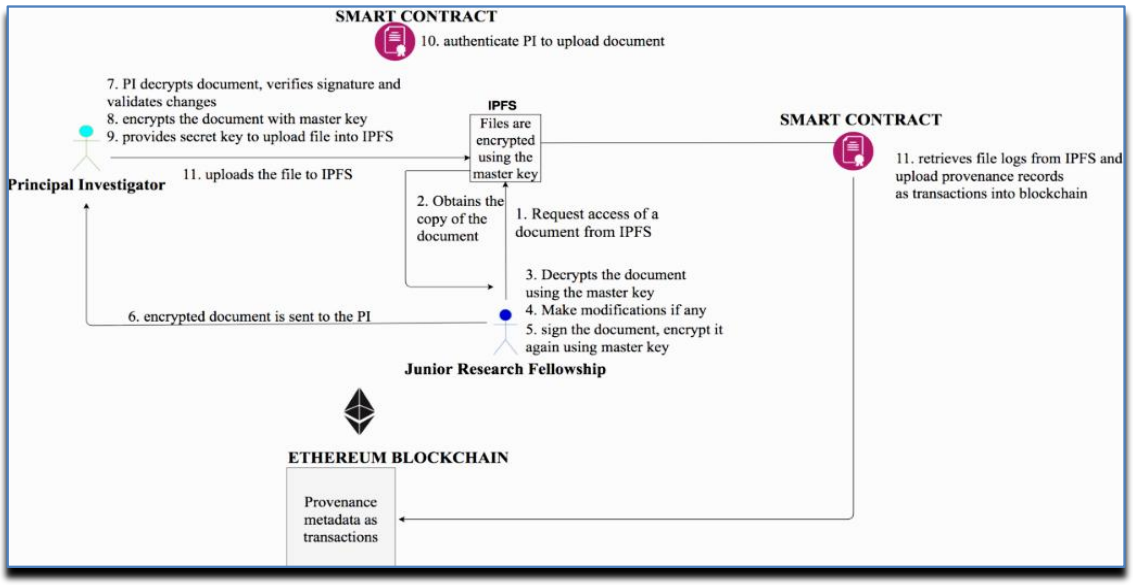

Fig. 1. Flow of the proposed framework

memorandum of understanding, funded project details, funding agency details, attendance records of the JRFs, and minutes of meeting. While considering the above documents, some can be accessed and modified by both the PI and the JRF and some can only be allowed to be modified by the PI. Hence, the access control policies must be defined according to the users using the system.

The framework proposed here can be divided into three main phases: User Registration and Authentication, Storage of documents and access control, Provenance Metadata information storage and retrieval for auditing purposes. An overall flow of the proposed framework is depicted in Fig. 1.

A. User Registration and Authentication

There are two different users involved in the system - Principal Investigator(PI) and Junior Research Fellow(JRF). There can be multiple PIs and JRFs who are using the system. So the registration of the users should be unique to avoid issues like impersonation. PI can be registered with details such as (PI id number, PI biometric details, PI name, password, secret question) while the JRF can be registered with details such as (JRF id number, JRF biometric details, id number of PI assigned, JRF name, password, secret question). These details can be stored in secure data stores which is either centralized or distributed.

Once the users are registered into the system, they can login into the system using their details. There are three steps of authentication in this system. In the first step users provide their id number and valid password. If it succeeds then they would have to pass a second layer of biometric authentication process where they should provide their biometric details. If that too succeeds then the third layer would be the secret question. Only if all these steps succeed, users can successfully login into the system.

If the user fails to succeed in any one of the steps, then the authentication would be unsuccessful. The three step authentication for secure login into the system is depicted in Fig. 2.

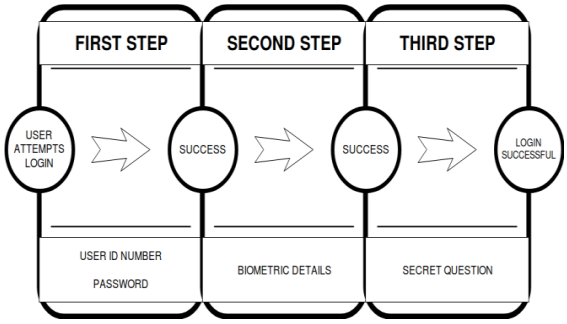

Fig. 2. Three step authentication process

B. Storage of documents and access control

The documents such as project reports, project funding details, memorandum of understanding, attendance records, and minutes of meeting are encrypted and stored in the IPFS. IPFS is a distributed file system that creates a unique hash of each document, and the other nodes on the network can access and view the files only if the unique hash of the file is known to them. In order to restrict access to particular nodes on the network, certain access control methods can be applied.

Two ways of access control can be applied: One way of restricting access is by implementing asymmetric encryption scheme through GnuPG [19].In this scheme, the users who have the key only can decrypt the document. Others cannot decrypt the document.

Even if the link of the document is provided to the users, the users can only decrypt the document when they have the key. So in the proposed system, this particular secret key is called as the master key. This master key is provided to the users who are allowed to access the system when they are registering with the system. Using this master key both the PI and JRF can access the IPFS file system and access the files. So, all the files are accessible by the users who are registered into the system. The user can login into the system and access the file, download it, decrypt it and then view it. But, an important restriction is applied here. Only the PI can upload and create documents on the IPFS network. Uploading and creating new documents in IPFS network is restricted for the JRF.

Consider the first scenario, where the JRF wants to access a particular document from IPFS. In this case, there is no problem since the JRF can obtain the link of the document, download it, and decrypt the document using the master key, and then view the document. JRF can do anything with the downloaded document such as modifying it or deleting it. But all these operations will be affected only in the local copy of the JRF's downloaded document. The original document in the IPFS remains unchanged.

Consider the second scenario, when the JRF wants to make changes to the original document in the IPFS. In this case, the JRF authenticates himself into the system, obtains the link of the document, download it, and then decrypt the document using the Master Key, view the document and make changes to it. But now this document has to be reflected in the IPFS File system. Only the PI can upload, create and modify the documents in the IPFS. So, the JRF makes changes to document, sign the document with his own digital signature and encrypt the document again with the master key and send it to the Principal Investigator. Now the Principal Investigator decrypts the document, verifies the signature and validates the changes. Then the document is again encrypted with the master key and then the PI authenticates into the system using his own secret key called as the derived key and then uploads the document to the IPFS. Smart contracts are written to authenticate PI using the derived key for uploading contents. This derived key is random in nature, so each time the PI wants to upload a document into IPFS there is a different secret key to be given to upload the modified document.

The derived key is usually derived from a password by using Password based key derivation function. A key derivation function (KDF) [18] is a function which derives one or more secret keys from another secret value such as a password or a master key using a Pseudo Random Function (PRF). A KDF is used for key strengthening and key stretching. There are many modern based key derivation functions. One such is the PBKDF2 [16] which is considered to be secure against brute force attacks (as of 2017) and another one is the simple HKDF which is a simple key derivation function [17]. The derived key can be obtained as follows:

Derived Key DK = KDF (PRF, Password, Salt Value, Number of iterations, Desired Length of the derived key).

Here, KDF is the key derivation function such as a keyed HMAC or simple HKDF, PRF is the Pseudo Random Function, Password is the secret password of the PI, and Salt Value is a sequence of random bits.

All the changes made to the document would be stored as different versions by the PI. If required, the PI can merge all these versions into a single version to get the final document. Since all the changes made are stored as different versions, the original document is safe and unaltered.

C. Provenance metadata storage and retrieval for auditing purposes

The next step is provenance metadata information retrieval and storage. Metadata here refers to the provenance data that has been collected from the IPFS file system. The provenance metadata information that has been collected from the IPFS file system can be: Name of the file, Hash value of the file, File creation time, File access time, IDs of the JRF who accessed the file, IDs of the PI who accessed the file. IPFS file logs can be collected for this purpose.

Consider the scenario where the JRF with id number say JRF_1 want to access a document A with hash value say HashA. The JRF obtains the document from IPFS, downloads it and decrypts it. Here the access time of the file accessed by the JRF would be recorded. Then JRF modifies the document local copy and request the PI for uploading it in the IPFS. The PI would then validate this request and then uploads it in the IPFS. This would again be recorded and now hash of the file is changed to HashB. The structure of the provenance data recorded is depicted in TABLE I.

This provenance data is very much necessary for further auditing purposes. The logs of the IPFS file system can be used to collect the provenance data of the documents stored. This provenance data information can be further embedded in ethereum blockchain as transactions using smart contracts which are built using Solidity Programming Language [13], Truffle [6] and Remix IDE [7].Algorithm 1 provides the glimpse of the smart contract to retrieve a file from IPFS, whose hash has been stored in blockchain using smart contracts.

Algorithm 1 Retrieve file from IPFS

```
procedure RETRIEVE FILE(A)
    filelocation = y
    hash = x
    if value x in A then
        Return y
    end if
end procedure
```

TABLE I
THE STRUCTURE OF A PROVENANCE RECORD

Record Id	File Creation Date	File Creation Time	Original Hash Value	File access date	File access time	Accessed by PI ID number	Accessed by JRF ID number	Hash of the modified file
R_1	12/03/2018	12.00 PM	HashA	13/03/2018	4.25 PM	--	JRF_1	HashA
R_2	13/03/2018	4.30 PM	HashA	13/03/2018	4.31 PM	PI_1	--	HashB

IV. ANALYSIS OF THE PROPOSED FRAMEWORK

The three step authentication method proposed in theframework,attempts to achieve the login mechanism of users in a secure way such that any malicious bypassing of the system will be unsuccessful. The access control mechanism ensures that, only the authorized person, i.e., the Principal Investigator, only can upload the modified documents in IPFS. This restricts the full access to IPFS for other unintended users. Even the JRFs are not given permissions for uploading content into IPFS.The mechanism provided for the JRFs who are requestingfor modification of the document, involves the use of encryption schemes and digital signatures, in a way that the signatures cannot be tampered. Further all the file logs collected from the IPFS file system are embedded into the blockchain as transactions. Any attempt to modify or tamper records on the blockchain network is not possible. The smart contracts in the proposed framework facilitate a trustful mechanism with no central authority involved. The system utilizes ethereum smart contracts. So a minimum amount of gas limit is required for each contract to be deployed. While deploying the smart contracts using MetaMask, there must be a minimum ETH and gas limit. Moreover the Truffle framework facilitates with default accounts each having a default of 100.00 ETH. This makes the deployment of smart contracts easier.

V. CONCLUSION AND FUTURE WORK

The framework proposed in this paper attempts to create a secure, tamper proof model for storage of research records in a distributed file system with no central point of control. Further the metadata information retrieved from the distributed file system is stored in the blockchain. This creates an immutable record of events on the blockchain, since blockchain is a distributed ledger technology which records all the transactions, which cannot be modified or altered. So, this avoids malicious modifications to the metadata information embedded on the blockchain. The framework proposed only involves two important users: Principal Investigator(PI) and Junior Research Fellow (JRF). This framework is feasible and can also be extended and implemented for other higher authorities above the PIs and the JRFs. Other mechanisms for access control can also be implemented along with the traditional encryption schemes suggested in this paper.

REFERENCES

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[2] Sankar, L. S., Sindhu, M., &Sethumadhavan, M. (2017, January). Survey of consensus protocols on blockchain applications. In *Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on* (pp. 1-5). IEEE.

[3] Benet, J. (2014). IPFS-content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561*.

[4] *An Introduction to IPFS – ConsenSys – Medium*. (2018). *Medium*. [Online]. Available:https://medium.com/@ConsenSys/an-introduction-to-ipfs-9bba4860abd0.

[5] *ethereum/wiki*. (2018). *GitHub*. [Online]. Available: https://github.com/ethereum/wiki/wiki/Patricia-Tree.

[6] *Truffle Suite - Your Ethereum Swiss Army Knife*. (2018). *Truffle Suite*. [Online]. Available:http://truffleframework.com/.

[7] *Remix - Solidity IDE*. (2018). *Remix.ethereum.org*. [Online]. Available: https://remix.ethereum.org/.

[8] *MetaMask*. (2018). *Metamask.io*. [Online]. Available:https://metamask.io/.

[9] Chen, Y., Li, H., Li, K., & Zhang, J. (2017, December). An improved P2P file system scheme based on IPFS and Blockchain. In *Big Data (Big Data), 2017 IEEE International Conference on* (pp. 2652-2657). IEEE.

[10] Ramachandran, A., &Kantarcioglu, D. (2017). Using Blockchain and smart contracts for secure data provenance management. *arXiv preprint arXiv:1709.10000*.

[11] Moreau, L., Freire, J., Futrelle, J., McGrath, R. E., Myers, J., & Paulson, P. (2008, June). The open provenance model: An overview. In *International Provenance and Annotation Workshop*(pp. 323-326). Springer, Berlin, Heidelberg.

[12] Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., &Njilla, L. (2017, May). Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (pp. 468-477). IEEE Press.

[13] *Solidity — Solidity 0.4.23 documentation*. (2018). *Solidity.readthedocs.io*. [Online]. Available: http://solidity.readthedocs.io/en/v0.4.23/.

[14] *Tierion: Blockchain Proof Engine | API*. (2018). *Tierion.com*. [Online]. Available: https://tierion.com/.

[15] *Chainpoint - Blockchain Proof Standard*. (2018). *Chainpoint.org*. [Online]. Available: https://chainpoint.org/.

[16] Moriarty, K., Kaliski, B., &Rusch, A. (2017). PKCS# 5: Password-Based Cryptography Specification Version 2.1.

[17] Krawczyk, H., & Extract, H. B. (2009). *and-expand key derivation function (HKDF). Request for comments*. RFC 5869.

[18] Chen, L. (2008). Recommendation for key derivation using pseudorandom functions. *NIST special publication*, *800*, 108.

[19] *The GNU Privacy Guard*. (2018). *Gnupg.org*. [Online]. Available: https://www.gnupg.org/.

[20] Feldman, P. (1987, October). A practical scheme for non-interactive verifiable secret sharing. In *Foundations of Computer Science, 1987., 28th Annual Symposium on* (pp. 427-438). IEEE.

[21] Lemieux, V. L. Blockchain and Distributed Ledgers as Trusted Recordkeeping Systems.