International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I1P239
Volume 70 Issue 1, 355-361, January, 2022

Design and Implementation of IPFS Enabled Security Framework for Multimedia Data Files

Vinodray Thumar[1], Dr. Saurabh Shah[2], Dr. Vipul Vekariya[3]

[1]*Research Scholar, Department of Computer Engineering, FTE, C U Shah University, Gujarat, India.*
[2]*Professor & Dean, School of Technology, GSFC University, Gujarat, India.*
[3]*Professor & Dean, Faculty of Engineering & Technology, Parul University, Gujarat, India.*

[1]vinod.thumar@gmail.com, [2]saurabh_er@rediffmail.com, [3]vekariya.vipul@gmail.com

Abstract *- Data security is an important aspect of the cloud storage environment. It is highly required to preserve very crucial user data in the cloud. Blockchain technology can become the prominent solution to maintain the privacy and security of data. The blockchain contains transactions in the form of a distributed database of records which are a public ledger for all digital activities or events that have been created and shared among participants. The details of all important data are stored in the form of transactions or blocks in the blockchain. Each activity on the community list is ensured by the consensus of the majority of participants in the process. A blockchain contains a specific and verified record of a single transaction. Confidential and important data files can be stored securely on decentralized storage using the proposed framework so that privacy and confidentiality will be maintained. Due to improved scalability, large data files can be stored in the blockchain. Transaction speed and block generation time are also focused in the proposed framework, which is irrevocable in nature due to the use of blockchain smart contracts and IPFS.*

Keywords – *Blockchain Technology, Data Security, Distributed Servers, Data Storage, IPFS.*

I. INTRODUCTION

Blockchain is generally a chain of blocks that holds all transaction details. It can generally be considered a public ledger. The chain grows with every new block added to it. Blockchain technology has a variety of important characteristics like persistency, auditability, decentralization and anonymity [1]. Hash values, digital signature [2] and distributed consensus are those few important technologies on which blockchain can work in the decentralized network. Due to its decentralized working manners, it is cost-effective and efficient [3]. If the user information is disclosed within the cloud computing environment, financial and psychological damages will occur because of the leak of users' sensitive data. The protection of the saving and transmission information, like confidentiality and integrity, within the cloud computing environment, is especially studied [4]. Blockchain may be a representative technology for guaranteeing anonymity. When the cloud computing environment and blockchain are used together, it can be a convenient service that gives stronger security [5][6]. User anonymity is often ensured if the blockchain methodology is employed once saving the user information within the cloud computing environment [7].

Smart contracts [8] are digital programs that are stored on a blockchain, and they run when preset conditions are satisfied. They usually are used to control the execution of an agreement so that all users are sure about the result of the execution process and there is no third party involvement. The process can be completed without wasting time. They will also monitor the progress of execution and direct further actions once conditions are satisfied. The purpose of smart contracts is the reduction of need in trustworthy intermediates, arbitrations and social control prices, fraud losses, furthermore because of the reduction of malicious and accidental exceptions [9].

The InterPlanetary File System (IPFS) is a general-purpose protocol that has hardly any storage restrictions and can serve all files, large or small. It took advantage of Bitcoin blockchain protocol and network infrastructure through which it stores data that is unalterable, removes redundant files and acquires details of the address to access storage nodes for exploring files over the network [10]. The larger files are split into smaller chunks which allow IPFS nodes to download files from hundreds of HTTP servers concurrently. The IPFS network becomes very suitable in design and distribution so that it can be an easily accessible network with the content of the Network Delivery. IPFS is a P2P network model which is used for file sharing. It is distributed and decentralized network across many network nodes [11]. This applies to all types of digital content, including data like photos, videos, distributed information, all operating systems as well as static websites.

IPFS [12] is one of the distributed classification systems which may connect a few computing nodes over the network with an equivalent structure of files, and it will manage all the files with their versions from time to time. At the protocol layer, IPFS has a specific property for content

addressing to identify the files. IPFS represents all the file contents in the form of the hash only. It will not represent the contents in the actual form which are stored on that server. The hash of files in IPFS begins with "Qm", and therefore the hash is truly a multi hash named distributed hash table (DHT). The contents are distributed on multiple nodes over the network. For example, the IPFS node can search for the content with hash QmYebHWdWStasXWZQiXuFacckKC33HTbicXPkdSi5Yfpz6 and therefore, the IPFS node can operate within the distributed hash table that nodes have the content. In the IPFS object, there are two different files with different file names but with the same contents that can have an equivalent hash value. In Ethereum, Merkle Patricia tree structure [13] may be emulated in the form of IPFS objects. In the Ethereum blockchain, huge fees are applicable if data in large quantity is to be stored.
So only the hashes of files are stored on it rather than a complete file. Moreover, there is a link to access that stored file in the form of a hash for that particular file on the IPFS [14].

II. RELATED WORK

Ethereum is one of the popular blockchains in the public domain. There is no limit in block size in Ethereum theoretically. Generally, the Ethereum blockchain is not used to store data because it is very expensive to store documents in Ethereum [15][16]. It is widely used to store financial transactions. There are some incidents in which some people tried to hack the Bitcoin blockchain to store some unexpected data apart from financial transactions. Efforts are still going on to find out many solutions like blockchain, which have been designed just to store data [17]. When any files related to Bitcoin transactions are stored on the Bitcoin blockchain, they are stored in the form of hash values of those files [18]. The size of the transaction file is of any size, but the block size of the bitcoin blockchain is up to 4 MB. A hash value of fixed length is generated of that file which is created by inserting a file with a mathematical algorithm [19].

Stroj [20] is a decentralized system and also one of the cryptocurrencies in the crypto market in which users are allowed to store data very securely and efficiently. This system uses transaction ledger encryption with the private or public key for its security. The primary function of Storj nodes is to sell resources to store and transfer data, and they earn Storj coins in return.

The author in [21] discusses a system based on blockchain for tracking the origin of digital assets. It is about the digital contents are to be converted into a binary file, and it can be stored in the form of hash values in the blockchain. The hash values are represented as owner identifiers. The ownership of digital assets can also be authenticated with verification of the integrity of those data assets. There is a centralized unit that is responsible for verifying the security of digital documents called SOC. SOC checks the authenticity of the digital assets. The main weakness of this system is the absence of decentralization because it operates through a centralized security unit SOC. Due to the utilization of a centralized server for data file storage, there are certain possible issues like lack of trust and threats of attacks.

The author in [22] describes a blockchain-enabled system for publishing online books in which the integrity of the digital document is indicated. In this system, authorship of books or files can be achieved by storing them in hash format with the name of the owner. The integrity can be verified by storing the file contents in the form of hash and timestamp of the block as pair.
But when anyone tries to alter the file contents, its hash will be affected. Due to the change in the hash value, the smart contract will not be able to access that file, so the result is that the file contents were modified. It can become a serious threat to security.

The authors in [23] propose a personal data management system that uses the concept of blockchain. It is assumed in this system that the owner of digital assets has full authority for their data. It is a data access control system that works virtually and automatically, and it can reduce the trust issues from third party access. A trust-based computing management platform can be constructed to develop a storage system based on blockchain. But lacking is that there is no discussion about the feasibility of storing larger files.

The author in [24] presents a model which uses the time-stamping technique to verify the availability of the digital document based on blockchain technology. In this model, the digital documents are submitted to the system, and it will generate a cryptographic hash. The entire contents of these documents are converted in the form of hash which represents the authenticity of the document. If anyone tries to modify the contents of the documents, then the hash value will change, which will differ from the existing hash generated with the previous process. Though this system recognize the authenticity of this digital document, the problem related to the ownership of these digital assets is still unsolved

Our proposed approach enables the use of blockchain smart contracts and IPFS together. In this approach, the digital contents or data files are stored on the IPFS. The contents of these digital assets are converted into hashes that can be stored on blockchain smart contracts. Hence, the authenticity of those documents can be achieved [25]. Those documents or data files are retrieved through the hash. If anyone tries to modify the content of the data files, the hash value related to those documents changes and it can be traced that the original content was modified and altered. Hence, the traceability of those data files can be achieved.

III. PROPOSED METHODOLOGY

A. Proposed Approach

The data files are stored in the blockchain in the form of blocks, which holds entire contents in hash values. The procedure to read the data and write the data to the blocks and form the blocks is described in the proposed algorithm. The main algorithm is created using the blockchain methodology, which is performed on the data transaction by the nodes in the network where the transaction is detected, and access control are done. By submitting a POLICY (u, s) package, active access enables users to change a set of permissions given to the application. All previously granted access rights are restored by posting an empty set. A user signing up for the service is perceived as posting a transaction for the first time with a new transaction ID. Reading and writing operations are also governed by data transactions. The details are only accessible to the user or application with CheckPolicy. It is observed in the algorithm that it is used the basic concept of accessing DHT as a standard hash table. In practice, these instructions cause an off-blockchain network message to be sent to DHT, which can be read or written.

$Procedure\ DATATX(pk_{sig}^{k}, m)$

$c,\ x_p, rw = Parse(m)$

$if\ Check\ Policy\ (pk_{sig}^{k}, x_p = True\ )\ then$

$pk_{sig}^{u,s}, pk_{sig}^{u,s}, POLICY_{u,s}$

$Parse(L|H(pk_{sig}^{u,s})])$

$a_{x_p} = H(pk_{sig}^{u,s} || x_p)$

$if\ rw = 0\ then\ \ \rightarrow rw = 1\ for\ read, 0\ for\ write$

$h_c = H(c)$

$L\left[a_{x_p}\right] \leftarrow L\left[a_{x_p}\right] \cup h_c$

$(DHT)ds[h_c] \leftarrow c$

$return\ h_c$

$else\ if\ c \in L\left[a_{x_p}\right]\ then$

$(DHT)return\ ds[h_c]$

$end\ if$

$end\ if$

$return\ \emptyset$

$end\ procedure$

B. Security Framework

As indicated in Figure – 1, a security framework using blockchain is described. Step by step process can be followed to create modules for different functions of the blockchain, like adding transactions, utilizing the micro-framework, and then executing the scripts on multiple machines to create a decentralized network. Building a simple user interface that interacts with the blockchain and stores information for any use case, details of transactions or data files. Data files uploaded in the blockchain are converted into hash values and stored over decentralized network storage.

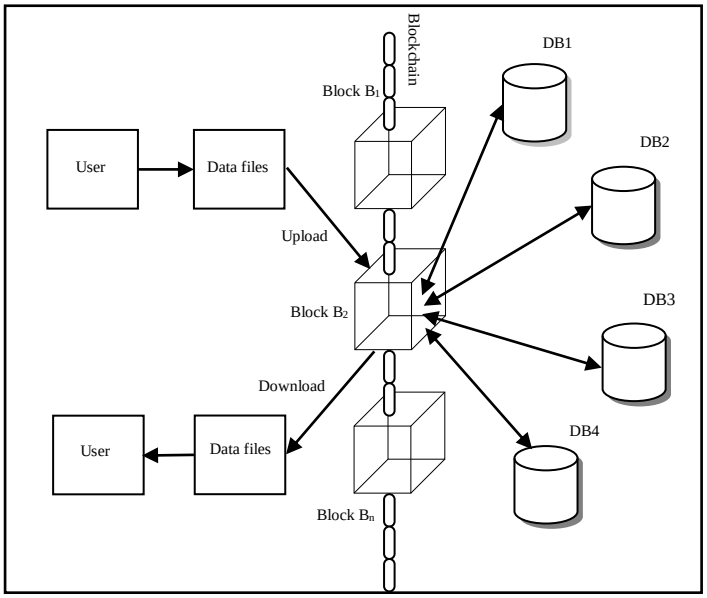

Fig 1. Secure framework using blockchain

C. System Implementation

The proposed security framework is implemented in python, which is used to implement the code and design the environment for this framework.

All uploaded data files on the blockchain are stored on distributed and decentralized storage servers in the form of blocks. Larger data files are distributed in small chunks and pieces of data that are arranged in a single block. Each block is assigned a two-character pseudo-code as its name in alphanumeric format. All blocks are chained together with a link address which will be used when it is required to retrieve that data file. Every file stored in that block is assigned SHA256 hash code as its name, and the entire contents of that file are also in the encrypted hash format, as indicated in Figure –2

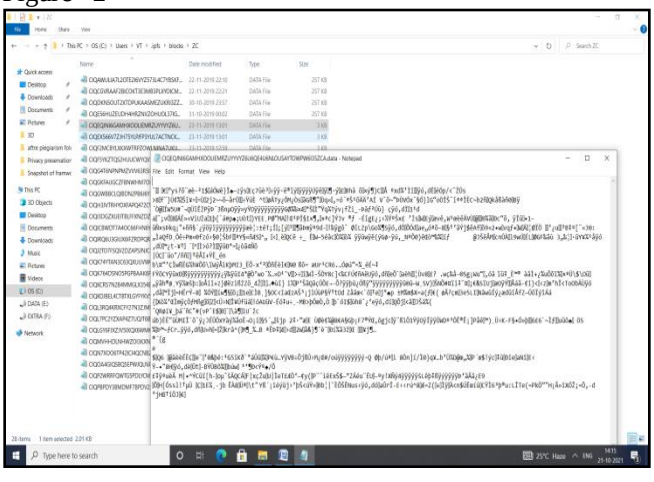

Fig 2. Contents of file in the form of hash stored on the blockchain

The block generation process is performed while the user uploads any data file on the blockchain. A list of multiple blocks with hash names is generated. The number of blocks will depend on the size of the data file. Each block contains a few chunks of the file. Time taken by each block to complete the generation process is called block generation time or transaction time. It can be visualized in Figure – 3. Total transaction time will depend on the total size of the file. It ranges from a few seconds to minutes.

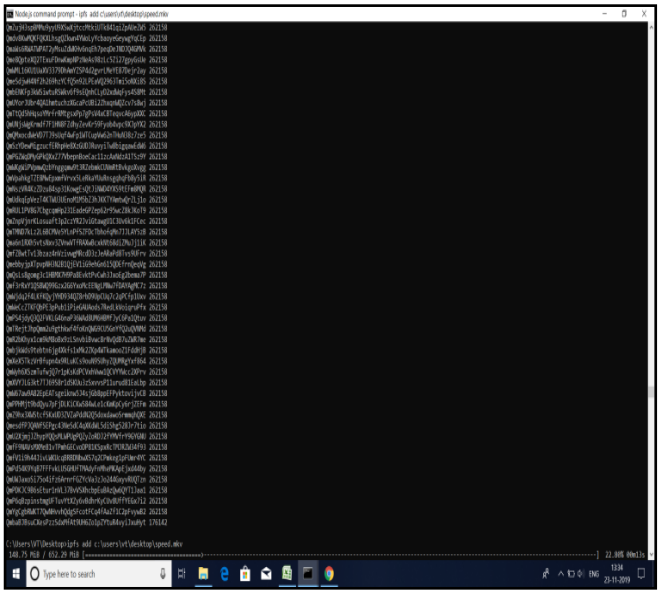

Fig 3. Evaluation of block generation/transaction time

IV. RESULTS AND EVALUATION

A. Datasets

In the test dataset, various multimedia files like text files, image files, audio files and video files are uploaded on the blockchain, and experimental results are obtained in the context of various parameters. Data files of various sizes are selected to evaluate the performance of the system. There are also variants in file formats in each category of the dataset. Table – I indicates various file formats of different multimedia file categories.

Table - I: Different file formats of multimedia files

Sr. No	File Type	File Formats
1	Text Files	.txt, .doc, .docx, .pdf
2	Image Files	.jpeg, .gif, .png
3	Audio Files	.mp3, .mp4, .wav
4	Video Files	.mp4, .wmv, .mov, .mkv

B. Evaluation Parameters

a) Scalability

Scalability means the capacity of blockchain to complete the number of transactions per second. Analytical solutions of average scalability in terms of transactions per second can be obtained with the mathematical model

$$SL = \sum_{i=1}^{n} \frac{(NB_i \; X \; NT_{x_i})}{T_{t_i}}$$

Where *SL* means total scalability for generated blocks for the specific data file. NB_i means total numbers of blocks generated during the transaction of the specific data file. NT_{x_i} This means an average number of transactions stored in a single block. While T_{t_i} Indicates total transaction time for the particular data file.

$$scalability\ (TPS) = \frac{SL}{N}$$

Where *scalability (TPS)* indicates average transactions per second of all the available data files at a particular time.

b) Block Generation Time

Block generation time is the time taken by the process to generate a single block which is then arranged onto the blockchain. To evaluate the block generation time, the numerical calculations can be processed with the mathematical model

$$BGT = \frac{TB}{T_t}$$

Where *BGT* indicates block generation time for the transaction of the particular data file, *TB* means a total number of blocks. T_t Indicates total transaction time for that particular data file.

c) Block Size

The block size of the proposed blockchain is 256 KB which is similar in size for all kinds of data files. Each block contains a number of transactions of a specific size within the block as its contents.

$$BS = \sum_{i=1}^{n} T_{x_i}$$

Where *BS* means block size in the transaction of specific data file while T_{x_i} means transaction of particular size stored in that block

C. Results

Experimental results of system testing on different data files which are uploaded on the blockchain are described in Table – II.

Table - II: Experimental results on different data files

No	File Name	Actual File Size	Total size on Block chain	Total Blocks	Block size	Total Transaction Time (sec)	Block Generation Time (sec)
1	File1.doc	2.49 MB	2.5 MB	10	256 KB	0.060	0.0060
2	File2.pdf	15.4 MB	15.5 MB	62	256 KB	0.380	0.0061
3	File3.jpg	7.48 MB	7.5 MB	30	256 KB	0.190	0.0063
4	File4.jpg	9.05 MB	9.25 MB	37	256 KB	0.220	0.0059
5	File5.mp3	6.29 MB	6.5 MB	26	256 KB	0.160	0.0062
6	File6.mp3	17.1 MB	17.25 MB	69	256 KB	0.430	0.0062
7	File7.mkv	652 MB	652.5 MB	2610	256 KB	16.970	0.0065
8	File8.mkv	1.13 GB	1.17 GB	4698	256 KB	25.840	0.0055

There are different kinds of blockchains like public blockchain as well as private blockchain. Various parameters like transaction speed, scalability, block generation time and maximum block size are evaluated with a comparison of the proposed blockchain with Bitcoin [26], KCC and Ripple [27]. Bitcoin is in the category of the public blockchain, while Ripple and KCC are in the category of private blockchain. Table – III indicates parameter wise comparison of results for the proposed blockchain.

Table - III: Comparison of results for different blockchains

No	Parameters	Bitcoin	Ripple	KCC	Proposed Blockchain
1	Transaction Speed	40 min	4 sec	8 sec	0.2 sec
2	Scalability	7 TPS	1500 TPS	1500 TPS	4000 TPS
3	Block Generation Time	10 min	15 sec	30 sec	0.006 sec
4	Maximum Block Size	1 MB	4 MB	4 MB	256 KB

D. Parameter wise comparative analysis of different blockchains

Performance of proposed security framework is evaluated with comprehensive analysis in terms of different parameters

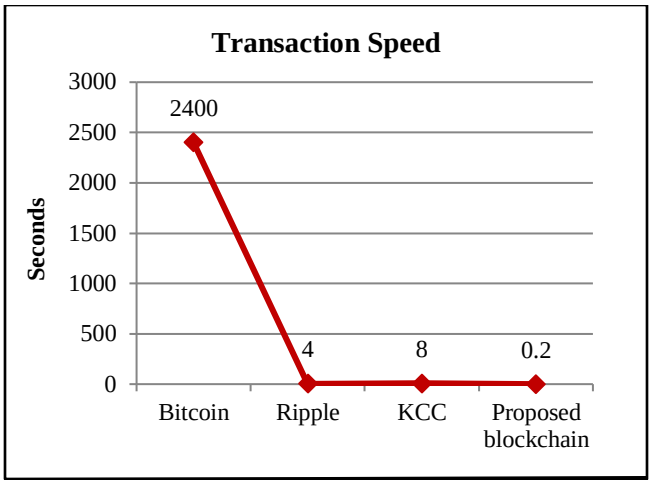

Fig 4. Comparative analysis of transaction speed for different blockchains

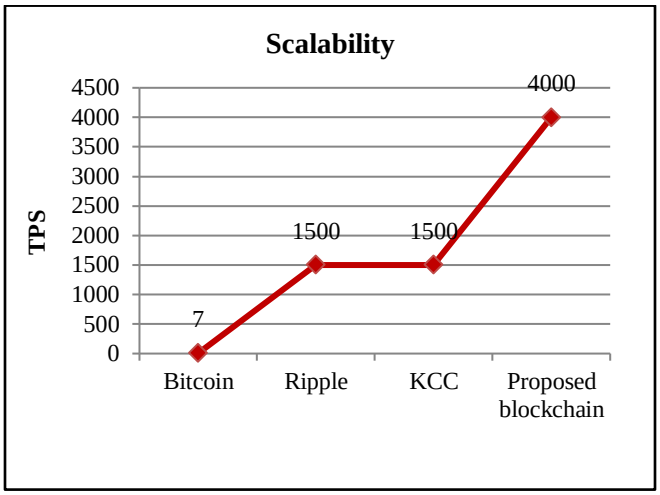

Fig 5. Comparative analysis of scalability for different blockchains

Like transaction speed, scalability, block generation time and block size, significant outcomes are obtained with this detailed technical system analysis. Figure – 4 describes a comparative analysis of transaction speed for different blockchains. The transaction speed of Bitcoin is 4 minutes, while the transaction speed of Ripple and KCC are 0.4 and 0.8 seconds, respectively. It is observed that the transaction speed of the proposed blockchain is 0.2 seconds which comparatively sound good in terms of processing of the data files during file upload and file download process. Figure – 5 represents a comparative analysis of scalability for different blockchains. The scalability of the Bitcoin blockchain is 7 TPS (Transactions per second), while the scalability of Ripple and KCC is 1500 TPS for each blockchain, and for the proposed blockchain, it is 4000 TPS which is significantly improved so that large data files can be stored on the blockchain.

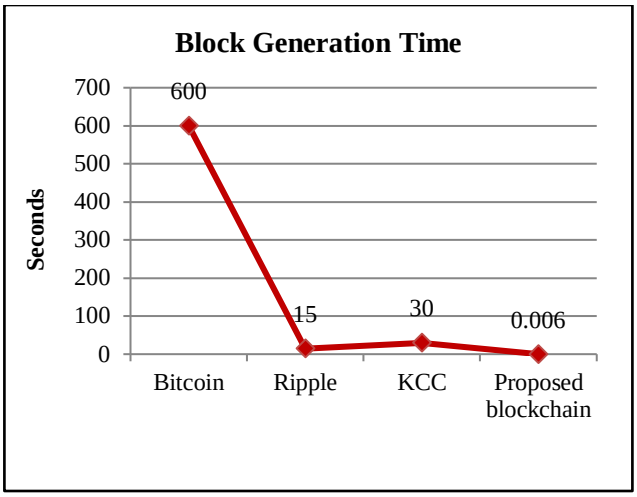

Fig 6. Comparative analysis of block generation time for different blockchains

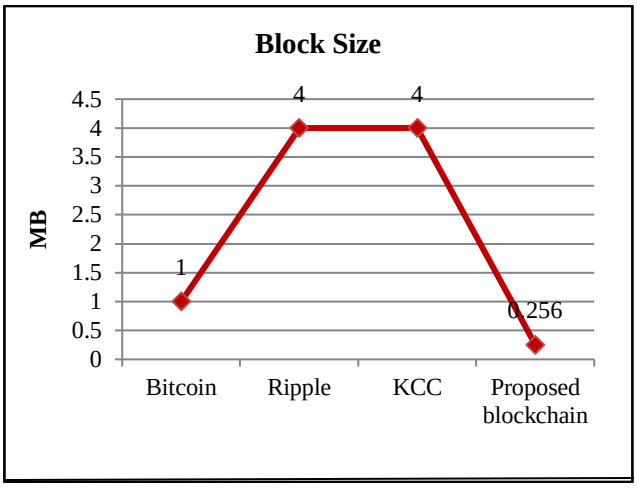

Fig 7. Comparative analysis of block size for different blockchains

As shown in Figure – 6, it represents a comparative analysis of block generation time for different blockchains. Block generation time for Bitcoin blockchain is 10 minutes, while block generation time for Ripple and KCC is 15 seconds and 30 seconds, respectively. It is analyzed that block generation time for the proposed blockchain is 0.006 seconds which is highly required to enhance the processing speed of data files transactions. Figure – 7 indicates the comparative analysis of block size for different blockchains. Block size for Bitcoin blockchain is 1 MB, while block size for Ripple and KCC is 4 MB for each blockchain. It is evaluated that the block size for the proposed blockchain is 256 KB. Due to smaller chunks of data files, it is very helpful to preserve the security of the data as all blocks are stored on decentralized storage servers over the network.

V. CONCLUSION

Apart from cryptocurrency, blockchain technology can also be used to focus on a data security perspective. Combining features of blockchain smart contract and IPFS together, security, scalability and feasibility of data storage can be enhanced. With this proposed framework, users do not need to be dependent on the privacy and confidentiality of data on the cloud service provider while storing data in the cloud or transferring data via a network. This framework will ensure the privacy and security of user data. Data in the form of blocks are distributed across decentralized storage, so they remain secured or untempered against any attacker. With enhanced scalability, larger data files are also to be stored on the blockchain. Due to high scalability, faster processing time and highly secured environment, this system can have a wide scope in various domains like Education, E-governance, UIDAI, Finance and Agriculture, in which data security is necessary.

REFERENCES

[1] Biryukov A., Khovratovich D., Pustogarov I. Deanonymisation of clients in bitcoin p2p network, In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. NY, USA (2014) 15–29.
[2] Johnson D., Menezes A., Vanstone S. The elliptic curve digital signature algorithm (ecdsa), International Journal of Information Security 1(1) (2001) 36–63
[3] Eyal I., Sirer EG. Majority is not enough: Bitcoin mining is Vulnerable, In: Proceedings of International Conference on Financial Cryptography and Data Security. Berlin, Heidelberg (2014) 436–454.
[4] Survey on blockchain technologies and related services, Tech. Report, NRI, METI, https://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf (2016)
[5] Zyskind G., Nathan O.: Decentralizing privacy: Using blockchain to protect personal data. In: Security and Privacy Workshops (SPW), 2015 IEEE. (2015) 180–184.
[6] Jin P., Jong P., Blockchain security in cloud computing: Use cases, Challenges and Solutions, Symmetry 2017, doi:10.3390 9(164) (2017)
[7] Hardjono T., Smith N. Cloud-based commissioning of constrained devices using permissioned blockchains, In Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security. ACM (2016) 29–36.
[8] Fries, Martin P., Paal, Boris. Smart Contracts, Mohr Siebeck, ISBN 978-3-16-156911-1 (2019).
[9] Buterin, V. A next-generation smart contract and decentralized application platform, white paper (2014).
[10] IPFS: a new peer-to-peer hypermedia protocol, https://ipfs.io (2016)
[11] Vincent T., Using IPFS for distributed file storage systems, https://medium.com/0xcode/using-ipfs-for-distributed-file-storage-systems-61226e07a6f (2020)
[12] Benet J., IPFS-content-addressed versioned, P2P filesystem, https://arxiv.org/pdf/1407.3561.pdf (2014)
[13] Ethereum-wiki, https://github.com/ethereum/wiki/wiki/Patricia-Tree. (2018)
[14] An Introduction to IPFS – ConsenSys – Medium. https://medium.com/@ConsenSys/anintroduction-to-ipfs 9bba4860abd0. (2018)
[15] Wood, G. Ethereum: A secure decentralised generalised transaction ledger, https://ethereum.github.io/yellowpaper/paper.pdf (2014)
[16] Zamfir, V. Introducing casper the friendly ghost Ethereum, https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost (2015)
[17] Ibm blockchain", http://www.ibm.com/blockchain/ (2016)
[18] Nakamoto S.: Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf (2008)

[19] Zyskind G., Nathan O., Decentralizing Privacy: Using blockchain to protect personal data. In: Security and Privacy Workshops (SPW) - 2015 IEEE. (2015) 180–184.
[20] Tim A., Kishore A. Cameron A. Storj: A Decentralized Cloud Storage Network Framework – white paper, https://www.storj.io/storj.pdf (2018)
[21] Ericsson Home Page, Ericsson White Papers –Industrial Blockchain and Data Integrity, https://www.ericsson.com/hyperscale/cloud-infrastructure/data-centric-security/data-integrity-assurance (2018).
[22] N. Prusty, Building Blockchain projects – Develop real-time practical DApps using Ethereum and Java Script, 1st edition, Packt Publishing Ltd, Birmingham, UK. (2017)
[23] G. Zyskind, O. Nathan, A Pentland, Decentralizing Privacy: Using Blockchain to protect personal data, Security and Privacy Workshops (SPW), IEEE Conference Proceedings San Jose, CA, USA, (2015) 180-184.
[24] P. Morgan, Using Blockchain Technology to Prove Existence of a Document, https://bravenewcoin.com/news/using-blockchain-technology-to-prove-existence-of-a-document (2018)
[25] M. Pors, Understanding the IPFS White Paper part 2, https://decentralized.blog/understanding-the-ipfs-white-paper-part-2.html (2017)
[26] Kenny L., The Blockchain Scalability Problem & the Race for Visa-Like Transaction Speed, https://towardsdatascience.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44 (2019)
[27] XRP: The Best Digital Asset for Global Payments, https://ripple.com/xrp/ (2019)