IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 8, August 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |

Evidence Storage system for large files using IPFS and Blockchain

Prathamesh Ingale[1], Om Mhatre[2]

U.G. Student, Department of Information Technology, B. K. Birla College of Arts, Science and Commerce (Autonomous), Kalyan, Maharashtra, India[1]

U.G. Student, Department of Information Technology, B. K. Birla College of Arts, Science and Commerce (Autonomous), Kalyan, Maharashtra, India[2]

ABSTRACT: Evidence collection is the first step for uncovering the truth of a crime. Evidence integrity is of utmost importance when it needs to be admissible in the court of law to prove a point. Even as the world moves towards a digital age, we discover that digital evidence is more fragile and easily tampering. In this paper, we propose a system model for an immutable evidence storage system for storing large digital evidence such as video files, audio files, images, reports, and other documents. In this model, we discuss the possible technologies that can be used to implement such a project and show an implementation of the system model using IPFS, ganache for blockchain, solidity for smart contract, python for the interface, and brownie framework as a middleware.

KEYWORDS: Blockchain, immutable storage, IPFS, Smart Contract, Encryption.

I. INTRODUCTION

In the 1980s, during the Justice Department's IranContra investigation Oliver North destroyed key documents [1]. During the early 2000s, Andersen destroyed Enron-related documents during an SEC inquiry and became the subject of a criminal investigation in its own right [2]. According to lawyers and judges, evidence tampering is not only limited to blockbuster events. Documents produced in evidence collection are regularly altered or suppressed. Investigators are regularly misled by the fabrication, falsity, or destruction of evidence. This may lead to letting accused persons go free and innocents getting punished. So, taking this into consideration we have proposed a model based on permissioned blockchain and Inter Planetary File System (IPFS).

The rest of the paper is organized as follows. Section 2 consists of related work. Section 3 reviews technologies that can be used in the proposed model. In Section 4, the system model is described in detail. In Section 5 we have demonstrated the system model with an implementation using some technologies shown in Section 3.

II. RELATED WORK

Miraz et al [3]discussed howblockchain can secure any type of transaction, whether it be human-to-human communications or machine-to-machine. They explained the working process of blockchain and surveyed FinTech applications of Blockchain. They also how blockchain is suitable in developing trust on the internet which was lacking in security. Various approaches were shown for securing data with the help of blockchain technology. Vora et al[4]showed how an efficient balance between the data privacy, and interaction of patients/providers to data was achieved using their BHEEM framework for medical record storage. Also, the probability of unauthorized use of the records was minimized with the help of encryption. Moinet et al. [5] proposed that using blockchain as an authentication database to store public keys, digital signatures, and peer information for a sensor network in machine-to-machine communication using a human-like knowledge-based trust (HKT). Michelin et al [6]expressed the importance of video evidence collected by surveillance cameras to support criminal investigations and how data integrity was the main issue in that case since someone can tamper with it. So, considering this they used Inter Planetary File System (IPFS) to solve that problem. Reegu et al [7]proposed an interoperable EHR system based on blockchain. This system achieved interoperability of the data along with secrecy, confidentiality, and integrity of data.This system achieved interoperability of the data along with secrecy, confidentiality, and integrity of data. Duc-Phong et al [8]proposed the use of a permissioned blockchain-based IoT forensics framework's architecture which was

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |

a cryptographic-based approach to reduce the privacy concern. The entire process i.e., evidence gathering, transmission, analysis, archiving, and disposal was proposed in this framework. Zhu et al[9] showed usage of identity-based cryptography that ensures that data integrity, privacy, and nonrepudiation can be achieved by Identity-based cryptography and electronic signature technology. Wang et al [10]have shown a new way for data integrity checking in large-scale IoT Cloud data storage by using Blockchain and Bilinear mapping (data integrity scheme).This was done by splitting data into shards and verifiable tags that have authentication metadata and then ensuring data integrity using multiple smart contracts. The data verification was done before the data gets stored on the cloud storage. The experiment was done using Hyperledger Fabric which outperformed blockchain and Merkle tree-based data integrity scheme and other schemes mentioned in the paper [10].

III. TECHNOLOGY REVIEW

Blockchain: Blockchain was proposed by Satoshi Nakamoto[11] in Blockchain is an underlying technology of Bitcoin which is a cryptocurrency developed by Satoshi Nakamoto in 2009. A blockchain is a growing list of records or data, which are called blocks. Blocks are linked to each other using the cryptographic hash value. Each block contains a cryptographic hash value of the previous block. Blocks also contain a timestamp, Merkle root, and transaction data. [12] Blockchain is immutable, decentralized, secure, and has distributed ledgers. In centralized forensic architecture, data integrity is a major issue that has to be addressed. Blockchain technology provides us decentralized architecture, so the control will be distributed among peers in the blockchain instead of giving it to a centralized server. Collective verification will be performed by verifying each transaction by all other nodes in the chain. Blockchain also provides tamper protection, so the data recorded in the blockchain cannot be modified or deleted.

Blockchain is of three types namely Public, Private, and Consortium Blockchain. Public Blockchain is one of the first and most widely used types of blockchain. Bitcoin (BTC) and Ethereum (ETH) are both Public blockchains. They are a network that anyone can join whenever they want. Private Blockchains are an implementation Public blockchain inside a restricted access network created especially for an organization or group of users. Consortium Blockchain is a type of blockchain in between Public and Private Blockchains which is usually used in fields where different roles like companies, governments, banks, organizations, etc. exist simultaneously.

Smart Contracts: Cryptographer, computer scientist, and lawyer Nick Szabo[13] first coined the term Smart Contract in 1990. They defined the term as a set of promises agreed in between multiple parties. Current advancements in Blockchain technologies make it possible for a computer program to handle the execution of a contract. According to IBM[14], smart contracts are simple programs that are stored on the blockchain which run when some previously decided conditions are met. [15] These smart contracts are used to automate the execution of agreements, transactions, transfer of assets, etc. They are buildup features of blockchain technology because they allow untrusted parties to create contract terms in the program and eliminate the requirement for a third party as a trusted party.

Inter Planetary File System (IPFS): The Inter Planetary File System (IPFS)[16] is a protocol used for storing and sharing data in a distributed file system present in a peer-to-peer network developed by Protocol Labs which also develops similar systems like IPLD and Filecoin. IPFS uses content-addressing which uniquely identifies each file in the network. IPFS also uses a decentralized system that allows users to host and receive the content in a similar manner to BitTorrent. Similar to BitTorrent, IPFS uses distributed hash tables (DHT) spread across multiple nodes which helps efficient coordination in access and lookup between nodes. Major advantages of using DHT are decentralization, scalability, and fault tolerance. Because of DHT, IPFS nodes do not require central coordination. With the help of IPFS within a blockchain transaction, we can place immutable links on the web which are in the network as long as they are still accessed regularly. IPFS allows access to content in a permissionless manner. IPFS has a special fork [17] since version 0.4.7 which allows private networks to be created using a special shared secret key to implement it for use of an organization or a group of organizations working together. This feature is stable currently but is not ready for a complete production environment.

Solidity: Solidity[15], [18] is an object-oriented, statically typed, high-level programming language used for writing smart contracts. It is used to create and execute smart contracts on various blockchain platforms which include most notably Ethereum. Solidity was developed by Christian Reitwiessner, Alex Beregzaszi, and several other Ethereum core contributors. The contracts written are compiled into program files. These program files run on the Ethereum Virtual Machine. To make the syntax familiar for developers, Solidity syntax is designed around ECMAScript syntax which is the syntax for JavaScript.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |

Brownie: Brownie [19] is a python based testing and development framework for smart contracts. It targets the Ethereum Virtual Machine. Its latest version has full support for Solidity smart contracts of version 0.4.22 and above and Vyper smart contract version of 0.1.0 and above. All testing done in brownie is done using pytest. Brownie is very easy to setup for developing Ethereum smart contracts. Brownie works primarily on a console for deploying smart contracts. For development, brownie has a GUI which gives the programmer opcode level control over the smart contract. For development, brownie gives a lot of information during debugging for when a transaction is reverted. Brownie has unit tests in python which give a stack trace analysis and evaluate for test coverage.

Truffle Suite: Truffle suite [20], [21] is the group of software such as truffle, ganache, Metamask, and drizzle used to create blockchain applications. It is a development and testing framework and works as an asset pipeline for blockchains using Ethereum Virtual Machine. It has inbuilt smart contracts compilation, linking deployment, and binary management. It also supports automated contract testing, rapid development, scriptable and extensible deployment, migrations framework, network management, interactive console for direct communication, and external script runner within the truffle environment.

Ganache: Ganache [22] is a personal blockchain created for agile Ethereum and Corda distributed application (DApp) development. It is used across the entire development cycle of a DApp, enabling the developer to develop, deploy and test your DApp. It can also be called an Ethereum simulator. It has a GUI as well as a CLI version for deploying and interacting with the personal blockchain.

Ethereum Mainnet: An Ethereum Mainnet [23] is an independent blockchain that allows its users to run programs on its network with all its technology and protocols. It is a live blockchain that has its cryptocurrency. Programmers use the testnet to troubleshoot and test new features and then the accepted features are rolled out in the Mainnet. This is largely focused on enterprise use of private blockchains.

Metamask: Metamask is part of the truffle suite used to interact with a DApp in a web browser. It is an extension for Chrome and Firefox browsers that connects to an Ethereum network without running a full Ethereum node on the browser machine. It can connect to the main Ethereum network, testnet or local ganache, or other blockchains that support Metamask

Hyperledger Fabric: Hyperlegder Fabric [24] is an open-source, modular, extensible, and permissioned blockchain. It introduces a new permissioned blockchain architecture aiming at resiliency, flexibility, scalability, and confidentiality. It overcomes the limitations of prior permissioned blockchains about consensus protocol, trust model, smart contract language, and many others. Fabric is written in the Go language. It uses the gRPC framework for communicating between clients, peers, and orderers. Programs based on it can be written in Node.js, Go as well as Java although go is the most well-supported language. The smart contract in Fabric is called chaincode and is written in Go language.

AES: AES is the Advanced Encryption Standard which is the subset of Rijndael block cipher[25] established by the US NIST in 2001[26]. NIST established 3 block sizes for encryption which are 128, 192, and 256 bits. AES encryption is based on a key of size equal to block size. AES is based on the principle known as substitution-permutation network. TOP SECRET level of encryption is done usually with 192-bit or 256-bit key lengths. This encryption is the most secure symmetrical encryption technique in its block size. Most programming languages have their implementations of encryption techniques like PyCryptoDome [27] for Python and crypto/aes[28] for Go.

IV. SYSTEM MODEL

Our system model consists of two entities:
IA (Investigating Agency): IA is the person or team of investigating personnel who are working on evidence collection.

JA (Judicial Agency): JA is the law authority that will review or interpret the evidence for further processing.
In the following diagram and implementation, we have not considered any blockchain integrity miners as they are not significant membersof this model. Also, we have assumed that we are implementing a robust authentication system and have a secure communication protocol for communication for purposes of demonstration.
Steps in the investigation:

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |

1. IA starts the smart contract with their node and initializes an editing user “Investigator” (adding files only) (for themselves) and a view-only user” Judiciary” (for JA).
2. Through the GUI, IA logs in through “Investigator” and uploads evidence and findings to the application.
3. The application splits the file into shards of 16 MB each.
4. The application encrypts each shard with AES Encryption and the encryption key is stored.
5. The application uploads each file into IPFS and hash is obtained for the shard and is stored.
6. The application calls addFile function with “Investigator” credentials on a smart contract which stores the file metadata i.e., a hash table with the encryption key, shard hash, and file names that get stored on the blockchain.
7. After upload, the application pins all hashes on any 3 nodes for redundancy.
8. After this, the application gives acknowledgment to IA that the document has been submitted.
9. After the investigation is complete, the IA locks all the documents for access from IA and gives access only to JA.
10. Through the GUI, JA logs in through “Judiciary” and obtains all files submitted as evidence.

Our work consists of the following algorithms.
File encrypt (S) => (key, ES): The file encryption algorithm takes in every shard (S) and outputs an encrypted shard (ES) and randomly generated encryption key.
Split File (F) => (S, M): The split file algorithm splits the (F) file into equal shards of N MB block size (S) and creates a manifest file(M) for the reassembly of the file.
File decrypt (ES, key) => (S): The file decryption algorithm takes in every encrypted shard (ES) and key and outputs respective shards.
Merge File (S, M) => (F): The merge file algorithm merges the shards using the manifest file to recreate the original file. Then the hash is verified by blockchain to be correct.
Main contribution: Our main contribution in this model is the feature of splitting files while transferring them over the internet so that every shard has the same file properties. This feature contribution will help a user to implement such a system to a trustless network and still work properly. This will also make sharing large blob files on the internet easier to share as there will be many nodes available. This will allow maximum throughput through the network and improve access times.
Why permissioned blockchain: The evidence storage system involves a lot of critical private and sensitive information for which we cannot let nodes join freely, therefore we cannot use Public Blockchain. In our use case, the evidence stored must be accessible to different levels of our investigating organization.
Access control: Only two parties get access to the files that too only one at a time. All access done by users can be logged easily as a logging transaction.

V. CASE STUDY

Initialization of environment

IPFS initialization: As we are working in an organized fashion, we will set up ganache client and IPFS daemons in private network configurations. For IPFS, we emulated different IPFS servers using a bridged docker network with multiple containers running go-ipfs images with environment variables for swarm key, ipfs output, ipfs staging, and private network environment variable. In production, these IPFS containers will be stored on different machines in the same network. Swarm key is a randomized hexadecimal key that is shared by all nodes on the ipfs network. It is the most important part of a private network. Using this swarm key, new nodes can be added to the ipfs network. It is similar to a password for the network. In our implementation, we have directly interacted with ipfs-cli to get the ipfs file hash of the file. Similar can be done using IPFS API for the programming language of your choosing.

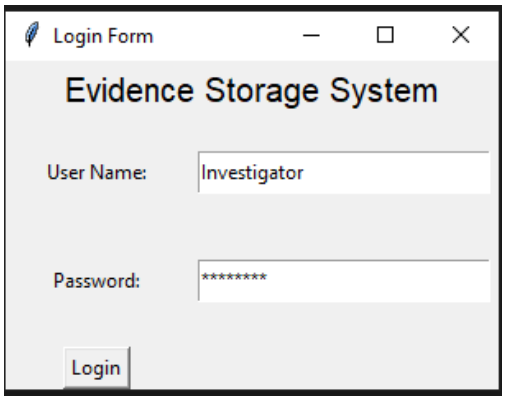
Brownie framework initialization: Now that we are have implemented the ipfs network, we will run our brownie script. Brownie will connect to a previously saved network configuration file that contains the port and IP address of the network and connects to it. It will launch ganache cli with 10 Ethereum accounts created and loaded in accounts variable. Now a GUI will appear which will allow the parties to upload and access files.

Uploading a file: When the Investigator wants to upload a file, they will enter a username, password to login, and the path of the file. The file will be copied to the input folder. Then the file will be split into shards of 16 MB and each shard is encrypted with a random key and then uploaded to IPFS directly. Then each shard’s IPFS hash, shard name, and the shard encryption key is passed on to the Ethereum blockchain in form of an array of each of them along with the original filename. This will be stored under a struct called file inside the smart contract.

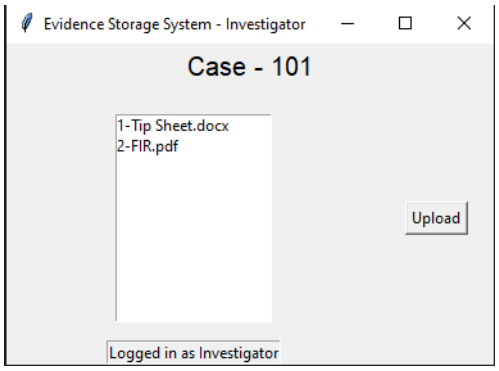
| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |



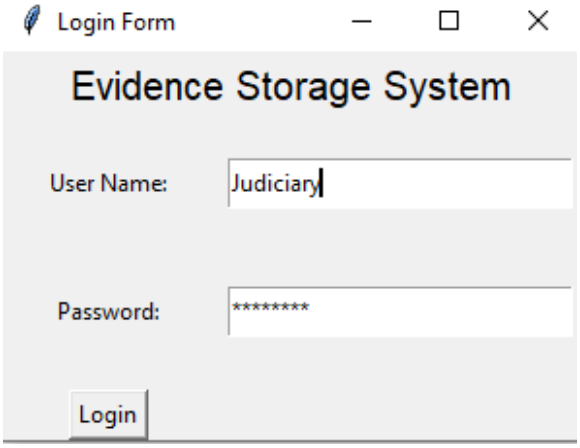
a. Investigator Login



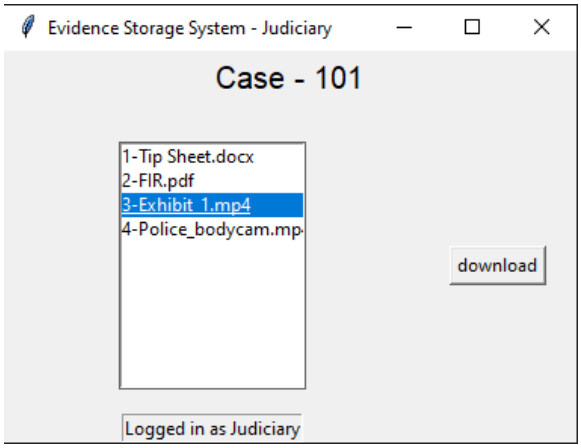
b. Casefile upload

Accessing the file:

To access a file, Judiciary enters the username and password to get the file. This information is passed to the getFile function of the FileStorage contract. The application downloads all shards from the IPFS by getting their address, shard name, and encryption key by executing the getFile function. Then each shard is decrypted using the key that came with it and then the files are merged to get the original file. This original file obtained will be stored to the user's desired location selected in GUI.



c. Judiciary Login



d. Judiciary case file review.

VI. SECURITY ANALYSIS

Avoiding single point of failure

In the proposed solution, as compared to other technologies, IPFS adopts redundancy, backup technology, Erasure coding, Proof of Replication, and Filecoin incentives to ensure data reliability and availability in a public network. At the same time, because IPFS runs in a peer-to-peer manner using DHT routing and BitTorrent technology, it offers higher data throughput at a lower cost. Also, IPFS pinning on multiple nodes helps overcome a single point of failure by making the file available for access regardless of its access frequency which is a determining factor for file serving in public IPFS. In a private network, IPFS pinning makes data available from more than one source which will increase the availability of data in a parallel way for faster throughput in a private network.

Security

Because of the data abstraction proposed in the system model, the only way to access the data of keys of shards, hashes of shards, and names of shards is through the use of proper credentials on the smart contract on the blockchain. The IPFS storage nodes work in a trustless manner. That is why the storage nodes can only see parts of ciphertext and cannot obtain information regarding them. Also, even if someone got access to some shards of a file, they would not be able to make sense of the data they collected because of encryption. Also, the merging of files depends on the name of

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |

the shard. So, somebody having access to all shards also won't be able to make sense of the data and because of encryption with AES, decryption without a key goes out of the question.

Future work

At present we have created a model that can help to store sensitive blob evidence data on IPFS in a secure way. In this very same way, distributed blob storage solutions can be created for high availability applications such as web video streaming, data feeds, surveillance storage, and many other places. IPFS makes it possible to store and serve data very efficiently and it has untapped potential because of it being HTTP only.

VII. CONCLUSION

The proposed system provides protection, security, and integrity required for the evidence to be permissible in front of the court of law. When a file is passed through the evidence storage application, it is obtained back without any problems in the file.The evidence storage system is created using IPFS for storage of files, Ethereum as blockchain, and Brownie as a deployment platform. Thus, a victorious result is obtained with the model described in the paper. It also helps us understand the close relationship possible between IPFS and blockchain applications.

ACKNOWLEDGMENT

We would like to thank Prof. Swapna Augustine Nikale, Department of Information Technology, B. K. Birla College, Kalyan for providing us with guidance throughout our research work.

REFERENCES

[1] C. W. Sanchirico, "Evidence Tampering," *Duke Law Journal*, vol. 53, no. 4, pp. 1215–1336, 2004.
[2] K. F. Brickey, "Andersen's Fall From Grace," vol. 81, p. 44.
[3] M. H. Miraz and M. Ali, "Applications of Blockchain Technology beyond Cryptocurrency," *Annals of Emerging Technologies in Computing (AETiC)*, vol. 2, no. 1, Art. no. 1, Jan. 2018, doi: 10.33166/AETiC.2018.01.001.
[4] J. Vora *et al.*, *BHEEM: A Blockchain-based Framework for Securing Electronic Health Records*. 2019. doi: 10.1109/GLOCOMW.2018.8644088.
[5] A. Moinet, B. Darties, and J.-L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," *arXiv:1706.01730 [cs]*, Jun. 2017, Accessed: Jul. 09, 2021. [Online]. Available: http://arxiv.org/abs/1706.01730
[6] R. A. Michelin, N. Ahmed, S. S. Kanhere, A. Seneviratne, and S. Jha, "Leveraging lightweight blockchain to establish data integrity for surveillance cameras," *arXiv:1912.11044 [cs]*, May 2020, Accessed: Jul. 09, 2021. [Online]. Available: http://arxiv.org/abs/1912.11044
[7] "Blockchain-Based Framework for Interoperable Electronic Health Record | Annals of the Romanian Society for Cell Biology." https://www.annalsofrscb.ro/index.php/journal/article/view/2166 (accessed Aug. 26, 2021).
[8] L. Đức Phong, M. H. Meng, L. Su, S. Yeo, and V. Thing, *BIFF: A Blockchain-based IoT Forensics Framework with Identity Privacy*. 2018, p. 2377. doi: 10.1109/TENCON.2018.8650434.
[9] X. Zhu and T. Fan, "Research on Application of Blockchain and Identity-Based Cryptography," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 252, p. 042095, Jul. 2019, doi: 10.1088/1755-1315/252/4/042095.
[10] H. Wang and J. Zhang, "Blockchain Based Data Integrity Verification for Large-Scale IoT Data," *IEEE Access*, vol. 7, pp. 164996–165006, 2019, doi: 10.1109/ACCESS.2019.2952635.
[11] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," p. 9.
[12] H. Sheth and J. Dattani, "Overview of Blockchain Technology," *Asian Journal For Convergence In Technology (AJCT) ISSN -2350-1146*, Apr. 2019, Accessed: Aug. 26, 2021. [Online]. Available: https://asianssr.org/index.php/ajct/article/view/728
[13] "First Monday: Formalizing and Securing Relationships on Public Networks." https://firstmonday.org/ojs/index.php/fm/article/download/548/469 (accessed Aug. 24, 2021).
[14] "What are smart contracts on blockchain," May 30, 2021. https://www.ibm.com/in-en/topics/smart-contracts (accessed Aug. 26, 2021).
[15] M. Wohrer and U. Zdun, "Smart contracts: security patterns in the ethereum ecosystem and solidity," in *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, Campobasso, Mar. 2018, pp. 2–8. doi: 10.1109/IWBOSE.2018.8327565.
[16] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *arXiv:1407.3561 [cs]*, Jul. 2014, Accessed: Aug. 26, 2021. [Online]. Available: http://arxiv.org/abs/1407.3561

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008149 |

[17] *go-ipfs*. IPFS, 2021. Accessed: Aug. 26, 2021. [Online]. Available: https://github.com/ipfs/go-ipfs/blob/e5b02b3de99ceaf3a5d913e93eb11e0139af4f8a/docs/experimental-features.md
[18] "GitHub - ethereum/solidity: Solidity, the Smart Contract Programming Language." https://github.com/ethereum/solidity (accessed Aug. 26, 2021).
[19] *Brownie*. Brownie, 2021. Accessed: Aug. 26, 2021. [Online]. Available: https://github.com/eth-brownie/brownie
[20] "Sweet Tools for Smart Contracts," *Truffle Suite*. https://trufflesuite.com (accessed Aug. 26, 2021).
[21] P. Hartel and M. van Staalduinen, "Truffle tests for free -- Replaying Ethereum smart contracts for transparency," *arXiv:1907.09208 [cs]*, Jul. 2019, Accessed: Aug. 26, 2021. [Online]. Available: http://arxiv.org/abs/1907.09208
[22] "Ganache," *Truffle Suite*. https://trufflesuite.com/ganache (accessed Aug. 26, 2021).
[23] "Enterprise on Ethereum Mainnet," *ethereum.org*. https://ethereum.org (accessed Aug. 26, 2021).
[24] E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, Apr. 2018, doi: 10.1145/3190508.3190538.
[25] J. Daemen, "The Rijndael Block Cipher," p. 47.
[26] "FIPS 197, Advanced Encryption Standard (AES)," p. 51.
[27] "pycryptodome/index.rst at master · Legrandin/pycryptodome," *GitHub*. https://github.com/Legrandin/pycryptodome (accessed Aug. 26, 2021).
[28] "aes package - crypto/aes - pkg.go.dev." https://pkg.go.dev/crypto/aes (accessed Aug. 26, 2021).

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details