Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

Securing Cloud Based Data Storage using Blockchain

Aishwarya Patil
Dept. of Computer Engineering, PCCOE
Pune, India

Swapnajit Patil
Dept. of Computer Engineering, PCCOE
Pune, India

Sachin Rokade
Dept. of Computer Engineering, PCCOE
Pune, India

Vijay Sharma
Dept. of Computer Engineering, PCCOE
Pune, India

Prof. G. B. Sambare
Dept. of Computer Engineering, PCCOE
Pune, India

***Abstract* - Nowadays, internet is the most common way used to share data around the globe. This sharing is backed by various cloud providers that allow customers to store & share data on the internet. But when it comes to privacy, cloud providers have consistently failed to make data 100% secure. Many data breaches, data piracy, hacking attacks have threatened the security mechanism of cloud providers. Though data stored by the customers should be 100% secure as it may contain private data which must be only accessible to the owner itself and some intended audience. So, it is very important to make this system more secure, so that data privacy & trust onto cloud providers can be maintained. With this paper, we introduce a system that leverages the security of cloud-based data onto the blockchain. It allows users to store data onto the cloud and provides a prominent access control mechanism that will ensure the privacy of data. Users will be able to share data with other people in a permissioned manner by sharing the link for document with the intended user. Logs of all the operations performed with the document will be available to the owner at any instance of time. This will ensure the actual ownership & privacy of the data. Any person or third-party will not be able to access document without valid permission. This will make existing cloud storage more secure & decrease the data breaches & several attacks.**

Keywords: Blockchain, Cloud Storage, Cloud Triggers, Smart Contract, Logs, Ethereum, Ledger, Data Privacy, Data Security, Solidity, Cloud Provider.

I. INTRODUCTION

Most organizations around the world create huge amounts of data through their day-to-day work. This data needs to be stored somewhere and should be easily accessible. This storage & easy access to huge data itself becomes a big problem for such organizations.[7] Cloud as an emerging technology provides a solution to this problem by allowing such organizations to store data on cloud storage and it can be easily accessed through the internet. This also resulted in a vast shift of organizations from on-premises to cloud. With with the adoption of cloud storage, organizations don't need to manage their data locally and also make this data available anytime anywhere through the internet as per the requirement of the user(public access/private access).

Though this seems to be very efficient from the organization's perspective, there are various risks associated with cloud storage. Maintaining the security, integrity & confidentiality of this data should have the highest priority while opting for cloud storage. Large cloud providers don't guarantee the security of data being stored. Whereas, there are continuous data breaches, data leakages happening with such cloud-based data. There are very few options available that could be used to ensure the security of data stored on the cloud.[13][17]

With the system proposed, we leverage the security of cloud-based data onto the blockchain. The results of the system promise to provide maximum security for the data stored on the cloud. The permission based access control makes the system more reliable & trustable, in turn making the data more secure. The results presented make it clear that the proposed system provides a better way to tackle above problems related to security of the data stored in the cloud.[4]

The logs accessible to the user ensure that each and every access operation to the data, let it be read or write or delete, are known to the actual owner of data. This preserves the integrity & intactness of the data. Also, it ensures that the permissioned user is only able to access the data. The logs will be responsible to give this information to the owner of the data. The logs also detect the malicious access to data thereby informing the owner about the unauthorised access. In short, the system with its all features provides a solution to various problems associated with cloud data storage.[8][9]

II. SYSTEM DEVELOPMENT

A. Architecture

Fig 1 depicts architecture for the system, it gives a short idea about the basic working and the technical flow of the system.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

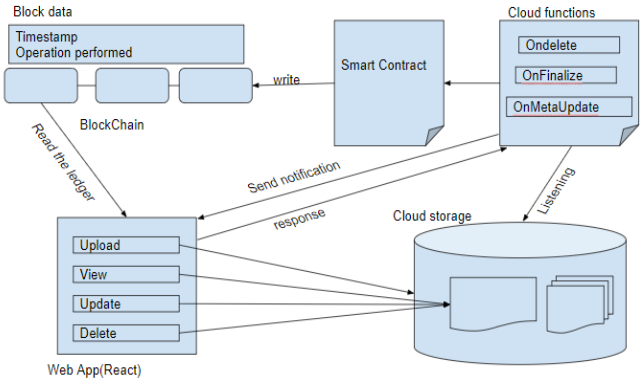

Fig-1:System Architecture

1. DApp(Web App)

Distributed App(Dapp) is an ethereum based distributed web application. We have used the React framework for the development of this Dapp. Using this dapp, the users will interact with our system, where users will have to register themselves and then login with the credentials used while registering(email, password). Only after successful login, users will be able to perform various operations related to documents like upload, view, delete, etc. Also, for their own documents as well, users will be able to share documents and grant/reject permission through dapp only.To check logs related to various operations performed on various files, users will be able to access that through this Dapp. In short, Dapp will facilitate all the communication of end users with our system.[6]

2. Cloud Storage

Cloud storage works as a core component of our system. All documents uploaded by the user are stored into the cloud. The documents will be uploaded from the local system to the cloud so that they can be globally accessed by authorized users through the internet. In our case, we'll be using Google Cloud Storage[8] as a cloud storage option. The documents will be securely stored onto the cloud & shareable links for the same will be available to the owner of documents. These available links users can share with others so that documents will be accessible to other authorized users. [7]

3. Cloud Storage Triggers

Storage triggers are an important part of our system, as they will be actively looking onto the documents whether any access attempt is made for the documents. In our case, Google Cloud Functions[8] will work as storage triggers that will keep a watch on files stored asynchronously. Whenever any access attempt will be made to any document stored, cloud functions will detect that and send a notification to the owner no matter whether access is done by authorized or unauthorized user. Then it will be upto the owner whether he wants to allow the operation or not. Based on his response, access will be given to the requester. Parallely, one smart contract will be executed through cloud functions that will invoke blockchain and all the details of operation like filename, access type, requester, access granted/rejected, etc. will be added to the blockchain. This information will work as log information for the owner.[7][19]

4. Blockchain

In our system, logs will be stored on the blockchain. As blockchain is tamper-proof, its immutability property will make the logs secure and no one can change & interfere with these logs. Every time access is made to any document, a transaction executes that adds log to the blockchain. So blockchain will have all the logs related to all the documents & all operations performed with these documents. Users can read the ledger from blockchain anytime, making the logs always available. Additionally, our application provides filters for filtering logs based on operation type, filename. E.g with filer users can view operations performed on specific files, or all read operations performed.

We have deployed our smart contract onto the Ropsten network which is a public network for ethereum smart contracts. And, smart contract is written in Solidity which is a popular language for writing smart contracts.[19]

So, with these four components as a pillar of our system, they form a solid architecture that makes our system realistic and implementable. Moreover, the data integrity & accessibility which are the most concerning factors from a security point of view are easily manageable with this architecture leading to the development of robust and secure application.

III. APPLICATION

A. Authentication & Login

Initially, the application requires users to be authenticated. The registration involves a simple form with email & password as login credentials. On registering successfully, users can login with the credentials(email & password). If by chance a user forgets the password, then the application provides a facility to change the password, that is also in a secure manner by verifying email first. The password reset link can be accessed through registered mail only.[3][5]

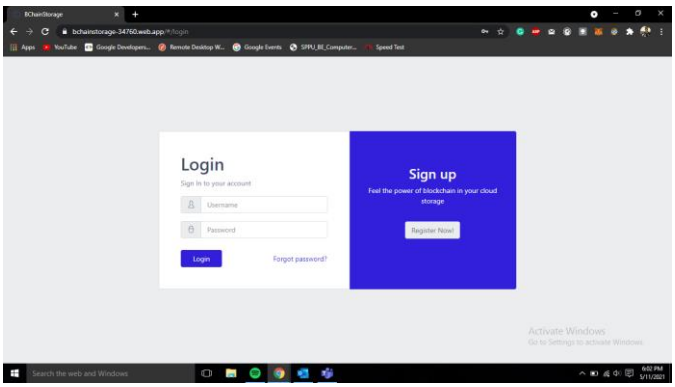

Fig-2: Landing Page

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

B. Home Page & Uploading Document

Once a user is logged in successfully, the initial page of our application is shown that has one button for uploading a new file, using which the user can upload the file to the cloud. Also, there is searchbox and view button available, with help of that user can access documents from other users by entering corresponding code of the file.[8]

As soon as the user selects a file, an alert is given and he has to confirm the notification whether he allows this operation or not. Also, he needs to confirm the transaction from metamask. Metamask is compulsory for this application as it accesses the blockchain. Once the transaction gets completed, the file is uploaded to the cloud. Finally, the log is stored on the blockchain for file creation.

C. Sharing Document

By clicking on the view icon, the user can view/download the file, this also requires the same approval from the user which will also trigger metamask transaction & the operation log details will be written to the blockchain. For sharing the document with other users or any other third-party, share button is available by clicking that it generates a code for file that can be shared with other users & file can be accessed using this code. Whereas, only allowed users can access the document as it will prompt for approval from the owner.

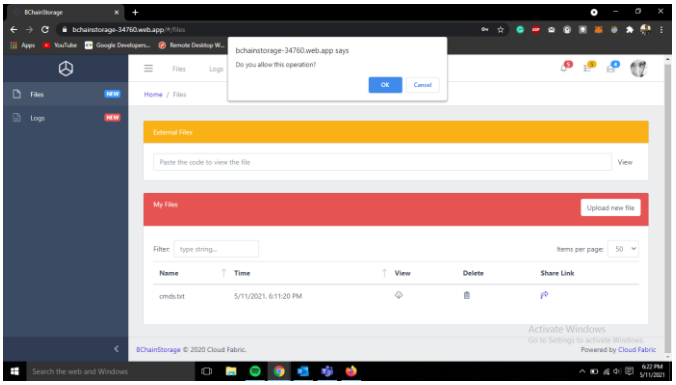

Fig-3: View & Share File

The application also has the facility to directly mail the code, so that it will remove the overhead of writing a message for sharing code. By only entering the email address, user can send code on a single click. Though for viewing & sharing files, metamask transaction need to be confirmed by the user so that log details can be added to the blockchain.[7]

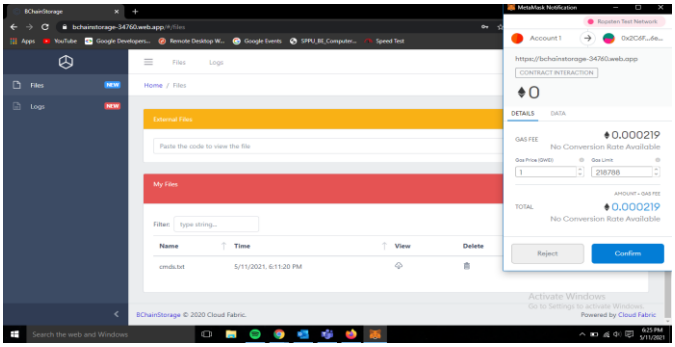

Fig-4: Metamask Transaction

At any instance of time, if any metamask transaction is rejected, then it will give a prompt to the user stating that the application is not able to add logs to blockchain & hence the corresponding operation won't be completed. This ensures that no operation is performed without writing logs to blockchain.[6][8]

D. Granting/Rejecting Permission

While a request is made by a second user which has valid code for file & right to access the file, a request is sent back to the owner. So, at the end it's up to the owner whether he still wants to give access to the file or not. Granting the permission will make the file available to the requester & if rejected then the requestor will not be able to access the file. In both cases permission is taken & a metamask transaction is conducted which adds respective logs to the blockchain.

E. Log Details

On clicking the Logs tab, the user will be able to access all the logs related to all the files which he owns. Logs are fetched from the blockchain that has all logs of previous operations stored while operations are performed. These logs consist of information like Filename, Operformed, Requester, Allowed(yes/no), Timestamp, etc. Additionally, there are textfields available for each field, which can be used for filtering logs for ease of access. The filtering can be done on the basis of any log field.[19]

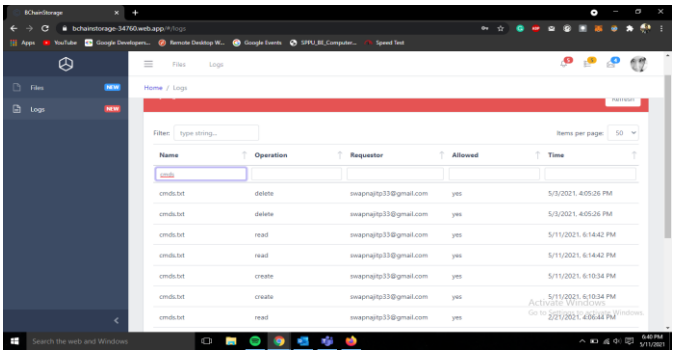

Fig-5: Log details

F. Smart contract used in Application

```
pragma solidity ^0.5.0;
pragma experimental ABIEncoderV2;

contract LogsStorage
{
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

```
struct UserDetails
{
   log[] logs;
}

struct log
{
   string documentName;
   string logContent;
   string logTime;
   string accessUserId;
   string logPermission;
}

mapping (string=>UserDetails) User;
event actual(log[]);
```

function writeLogsOnBlockchain(string memory userID, string memory logData,string memory logTime, string memory documentName, string memory accessUserId, string memory logPermission) public

```
{

   log memory l;
   l.logContent=logData;
   l.logTime=logTime;
   l.documentName=documentName;
   l.accessUserId=accessUserId;
   l.logPermission=logPermission;

   User[userID].logs.push(l);

}

function getLogsFromBlockchain( string memory userID) public  returns(log[] memory)
{

  uint numberOfLogs=User[userID].logs.length;
  log[] memory actualLogs=new log[](numberOfLogs);

   for (uint i = 0; i < numberOfLogs; i++) {
     log storage people = User[userID].logs[i];
     actualLogs[i] = people;
    }

   emit actual(actualLogs);
}
}
```

In the Smart contract mentioned above, there is a structure UserDetails defined for storing all the logs which contain an array of logs. logs array will store all the information like the Name of the document, type of operation(read/write/delete), the requester of the operation, permission(Yes/no), timestamp, etc. for every log. Whenever any user performs any operation on the document writeLogsOnBlockchain function will be called and will write logs for that user on Blockchain. When any user wants to see logs for all his documents, the getLogsFromBlockchain function will be called, and all the logs will be displayed to that user. Users can sort the logs based on any parameter after getting them from Blockchain.[1][15]

The Smart contract is deployed on the Ropsten test network. It can be accessed by searching the 0x2C6F74236A22D45E67d308353A5c05a5A84F6e8E address on the Ropsten test network Etherscan. Below is the list of all the recent transactions performed by different accounts on the contract.

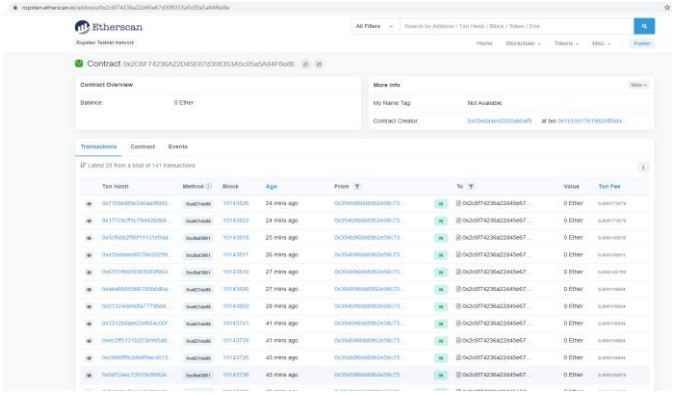

Fig-6: Smart Contract Transactions

IV. COMPARATIVE ANALYSIS

Existing Systems	Proposed system
There are Many theories that have proposed the use of encrypted documents	There is no overhead of encryption and decryption
Cryptography is involved and it requires secure channel for sharing of various keys	There is no overhead of sharing keys as Cryptography is not Involved
There is no Involvement of Blockchain	The benefit of blockchain ledger, Consensus and its tamper proof nature is used for achieving more security
There are no logs generated and stored for every operation performed on each document	Here Logs are stored on Blockchain for every document and User can access these Logs stored on Blockchain
Cloud providers have sole right on document storage. He might perform unsecure operations on the document	As cloud database triggers are involved cloud provider has to take permission from user before doing any operation on document
The shared document links can be shared further to anyone and User is unaware of who is using the link	If a document link is shared with anyone other than the trusted entity, it will notify and request permission from the owner of the document
If User's account is hacked, he will lose full control of his account	Here even if the account is hacked the owner of the document will be notified on his mobile before any unethical operation is performed on the document

TABLE I COMPARISON OF EXISTING SYSTEMS WITH PROPOSED SYSTEM

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

V. CONCLUSION

We have implemented a functional prototype of the system & demonstrated its working concerning added security provided on top of cloud providers. The average time required for encrypting the document using the current system that uses AES256 for encryption is 0.81 sec to encrypt input sizes between 1-10 MB and, for decryption is 0.755s to decrypt input sizes between 1-10 MB. The memory usage by AES256 is 32.5KB, energy consumed is 1.2mJ and latency is 0.019 MBPS. The time required to generate links of the document using SHA256 is 480ms for 10MB of data, Power consumed is 14.8 Wh and, latency is 62ms. In our System encryption, decryption, and generating the hash for uniquely identifying the documents in Blockchain is not necessary as documents are stored as it is with the mere addition of triggers and, we use User ID to uniquely identify the documents on the Blockchain. Our system will save the time mentioned above for generating the links, encrypting, and decrypting the documents stored on the cloud. It will also save the memory space and energy required for encryption and decryption. It will allow users to refer to tamper-proof logs stored on Blockchain for all the operations and share the data securely using two-factor authentication in which the first step is sharing the link with the trusted entity. And to prevent unauthorized access, the second is approving the notification for granting permission. This makes our system cheaper, secure and time-efficient.

REFERENCES

[1] Maximilian Wöhrer, Uwe Zdun, " Smart contracts: Security patterns in the ethereum ecosystem and solidity", International Workshop on Blockchain Oriented Software Engineering (IWBOSE) ,IEEE, 2018.
[2] Qiwu Zou, Yuzhe Tang, Ju Chen, Kai Li, Charles A. Kamhoua, Kevin Kwiat, Laurent Njilla, "ChainFS: Blockchain-Secured Cloud Storage", IEEE 11th International Conference on Cloud Computing (CLOUD), 2018.
[3] Fran Casino, Thomas K. Dasaklis, Constantinos Patsakis,"A systematic literature review of blockchain-based applications:Current status,classification and open issues" , Elsevier, 2018
[4] R.Gowthami Saranya, A.Kousalya,"A comparative analysis of security algorithms using cryptographic techniques in cloud computing" ,IEEE,2017.
[5] Ilya Sukhodolskiy, Sergey Zapechnikov, "A Blockchain Based Access Control System for Cloud Storage," IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), 2018.
[6] Shubham Desai, Rahul Shelke, Omkar Deshmukh, Harish Choudhary, Prof. S. S. Sambhare "Blockchain Based Secure Data Storage and Access Control System using Cloud" IEEE - ICCUBEA 2019.
[7] Gurudatt Kulkarni, Rani Waghmare, Rajnikant Palwe, Vidya Waykule, Hemant Bankar, Kundlik Koli, "Cloud Storage Architecture", IEEE 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2012.
[8] Google Cloud Platform Documentation: https://cloud.google.com/docs
[9] AWS Documentation: https://docs.aws.amazon.com/
[10] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, Qiaoyan Wen,"A Survey on the Security of Blockchain Systems",beijing university China, 2018.
[11] Rongzhi Wang, "Research on data security technology based on cloud storage", 13th Global Congress on Manufacturing and Management, GCMM, 2016.
[12] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends", IEEE 6th International Congress on Big Data, 2017.
[13] Julija Golosova et.al. "The Advantages and Disadvantages of Blockchain Technology", IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), 2018.
[14] Mr Anup R. Nimje et.al. "Blockchain Attribute Based Encryption Techniques in Cloud Computing Security : An Overview " IJCTT Volume 4 ,Issue 3-2013
[15] Sangsuree Vasupongayya -"Blockchain-Based AccessControl Model to Preserve Privacy for Personal Health Record Systems" , Research Article, 2019
[16] Naresh vurukonda, B.Thirumala Rao, -"A Study on Data Storage Security Issues in Cloud Computing", 2nd International Conference on Intelligent Computing, Communication & Convergence , (ICCC-2016), 2018. [17] Guang Chen, Bing Xu1, Manli Lu1 and Nian-Shing Chen, -"Exploring blockchain technology and its potential applications", [Elsevier] 2018
[17] Jin Ho Park, -"Blockchain Security in Cloud Computing: Use Cases, Challenges.", Seoul National University of Science and Technology, SoongSil University,Korea;Research Article,2017
[18] Zibin Zheng1, -"An Overview of Blockchain Technology: Architecture, Consensus", National Engineering research center of China, [IEEE] 2017