SECURE AND DECENTRALIZED FILE TRANSFER APPLICATION USING BLOCKCHAIN

SriBalaji[1], Vignesh Mohan[2], Soundarya[3]
Chennai, India

ABSTRACT

Growth in blockchain technology has been witnessed through the development of bitcoins and another important application called storj[2] which involves the concept of a distributed cloud storage. A more efficient application would be to enable file sharing through the concept of Blockchain. This would help in reducing the two step process of uploading a file to the drive and downloading it from the same to a single step process of just transferring it from a sender to a receiver in a Blockchain network. Even though there are several applications which provide file sharing, it cannot match the one that is based on Blockchain technology in terms of security. Our focus is to enable a secured file sharing application by using a private Blockchain network so that it can be used within small organizations. A greater level of security is achieved by applying some critical algorithms from the area of cryptography to strongly encrypt the file thereby making sure that none other than the receiver can gain access to the file.

Categories and Subject Descriptors

D.3.3 [Organization and Design]: Peer to peer, Decentralized

E.1 [Data structures]: Merkle tree data structures, Round robin mining.

Keywords: Block chain, Decentralized, File transfer, P2P, Anonymous.

1. Introduction

The Blockchain is an advanced distributed database system that maintains a continuous-growing of record blocks secured from tampering. Each block contains a timestamp and a link to a previous block, in a Merkle Tree Structure. The decentralized cloud is the advanced and next level cloud where the difficulties of normal centralized cloud are considered and taken care of.Normally Cloud storage is centralized where are all data are stored on a single server or a computer so that if the server is hacked or broken, the confidential data can be accessed thereby interrupting the service.. Also trusting the third party service is not recommended for confidential file transfer such as bank transactions, military secrets, etc., Normally military secrets are shared using TOR(The onion Routing), but the tor network is very slow, since it relies on many servers in various parts of the world, it can be hacked using government support. But by constructing a private blockchain network for file sharing, hacking into the blockchain network becomes very hard to an extent that even the network administrator cannot intercept or tamper any part of the file . Blockchain is a new and emerging technology and it mainly deals with the Cryptocurrency transactions where the whole transactions are fully secured and anonymous. The common example of public blockchain is bitcoins. In this project, instead of using Bitcoins, we are transferring files so that even the large files can be sent from one node to another node without uploading to any third party cloud.

1.1 Merkle Tree Data Structure

Merkle tree data structure uses a hash value to verify the message or data transferred between two nodes. It is very important in a P2P networks where we rely on unknown nodes. Consider if there are 4 messages namely 'a', 'b' , 'c' & 'd' ,each message is hashed individually, then hashed values of a and b is combined into "ab", and the hashed values of c and d is combined into "cd".

Further these are combined into "abcd" which is top most root hash.

Any changes in a single message also results in wrong hash values, and these hash values are compared with hash values of other nodes. If the values are not same, it assumes that some tampering has taken place and the transaction will not get confirmed. This maintains consistency as well as security in P2P networks.

1.2 Multichain Framework

Multichain[2] is an open source framework which allows user to deploy private blockchain for any organization. MultiChain supports Windows, Linux and Mac servers and provides a simple API and Command Line interface.

Multichain solves the related problems of mining, privacy and openness via integrated management of user permissions. Multichain is permission based private blockchain. Using Multichain framework any number of nodes can join to form a network.

1.3 Miners in Private Blockchain

Miners play an important role in blockchain and they are responsible for creating blocks at each node. There will be two set of block data out of which one is temporary and the other is permanent. The temporary blocks are created immediately when the transaction is initiated. Once the mining is done, the temporary block gets linked to permanent block which is the actual block used by all the nodes. Since blockchain built is private, mining need not be very complex and hard like the one performed for bitcoins.

The first block is called the "genesis" block which receives all the permissions initially. The administrator has to grant the permissions to the other users joining the network. Any node in a network can become a miner by requesting for permission from the admin but we have set up a network that provides miner permission for all the nodes present in the network and to any new node which joins the network in future so that the transaction can be confirmed quickly and the files can be received immediately. The mining concept has been implemented in a round-robin fashion .

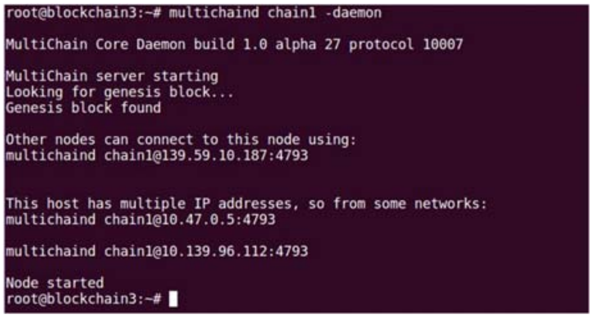
3. Deployment Scenarios

Consider a scenario for creating a private blockchain for a educational organization. The main condition for a file transfer to take place here is that both the sender and the receiver should have multichain framework installed in it.

3.1 Environment Setup

Consider that the private blockchain for an educational organization is created initially with 4 nodes. One is the local host itself, and the other nodes are three instances bought on AWS and set up in various locations such as Bangalore, US and Europe. With this, we have created a private blockchain network by connecting the instance using multichain framework running across the country. It is fully hardware independent and there is no need for any custom or specialized hardware to work with it. Normal laptop or PC is enough to conduct the test. It also runs in various cross platforms and supports different operating systems such as Linux, OSX and Windows.

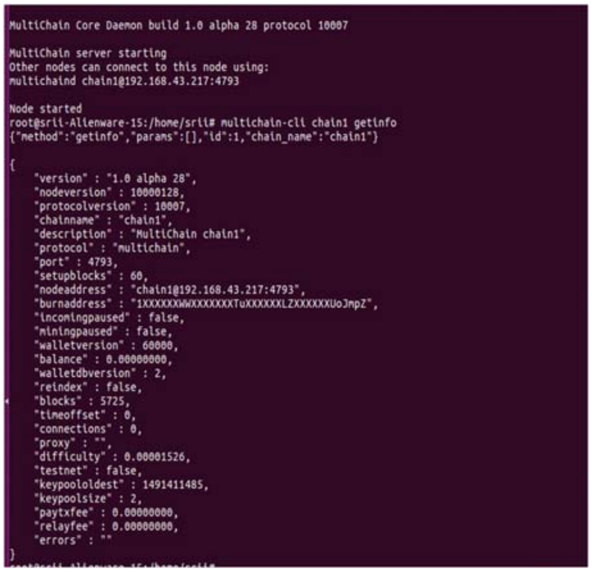
3.1.1 Connecting an instance to a private blockchain



```
root@blockchain3:~# multichaind chain1 -daemon

MultiChain Core Daemon build 1.0 alpha 27 protocol 10007

MultiChain server starting
Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind chain1@139.59.10.187:4793


This host has multiple IP addresses, so from some networks:
multichaind chain1@10.47.0.5:4793

multichaind chain1@10.139.96.112:4793

Node started
root@blockchain3:~# ▌
```

Figure 1. Starting the blockchain service

Once the "genesis" block is found the multichain framework on that particular node starts.



```
MultiChain Core Daemon build 1.0 alpha 28 protocol 10007

MultiChain server starting
Other nodes can connect to this node using:
multichaind chain1@192.168.43.217:4793

Node started
root@srii-Alienware-15:/home/srii# multichain-cli chain1 getinfo
{"method":"getinfo","params":[],"id":1,"chain_name":"chain1"}

{
    "version" : "1.0 alpha 28",
    "nodeversion" : 10000128,
    "protocolversion" : 10007,
    "chainname" : "chain1",
    "description" : "MultiChain chain1",
    "protocol" : "multichain",
    "port" : 4793,
    "setupblocks" : 60,
    "nodeaddress" : "chain1@192.168.43.217:4793",
    "burnaddress" : "1XXXXXXWWXXXXXXXTuXXXXXXLZXXXXXXUoJmpZ",
    "incomingpaused" : false,
    "miningpaused" : false,
    "walletversion" : 60000,
    "balance" : 0.00000000,
    "walletdbversion" : 2,
    "reindex" : false,
    "blocks" : 5725,
    "timeoffset" : 0,
    "connections" : 0,
    "proxy" : "",
    "difficulty" : 0.00001526,
    "testnet" : false,
    "keypoololdest" : 1491411485,
    "keypoolsize" : 2,
    "paytxfee" : 0.00000000,
    "relayfee" : 0.00000000,
    "errors" : ""
}
```

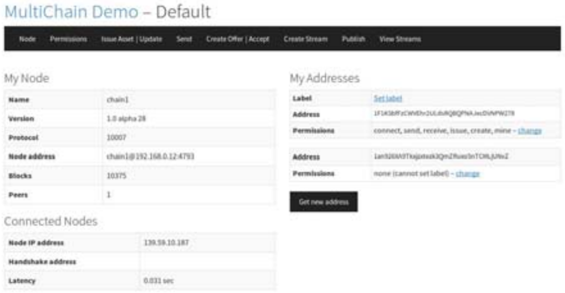
Figure 2.Chain information

This shows the number of nodes connected and the version of multichain, the number of blocks created and a masked burn address.

3.1.2 Private Blockchain web Explorer

As a bitcoin explorer we also have an explorer which records all the transactions made in our private blockchain.

The image below shows the list of nodes connected with its address. Also the address of the current node.



MultiChain Demo – Default

My Node

Name	chain1
Version	1.0 alpha 28
Protocol	10007
Node address	chain1@192.168.0.12:4793
Blocks	10375
Peers	1

Connected Nodes

Node IP address	139.59.10.187
Handshake address	
Latency	0.031 sec

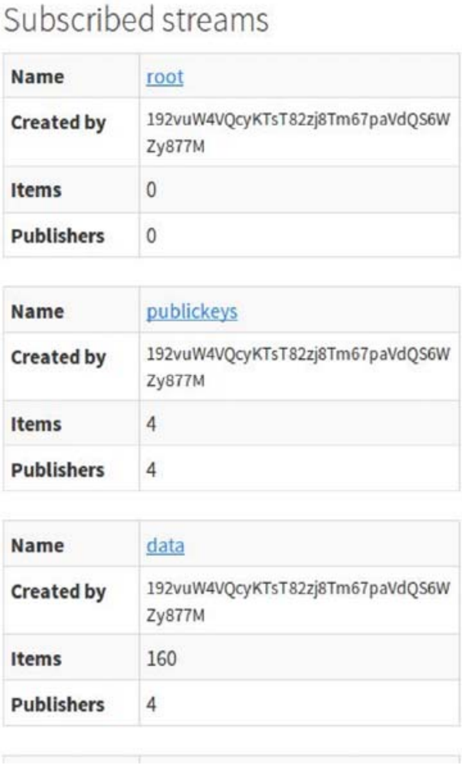
My Addresses

Label	Set label
Address	1F1KSbfFzCWVEhr2ULdsRQBQPNAJecDVNPW278
Permissions	connect, send, receive, issue, create, mine – change

Address	[illegible]
Permissions	none (cannot set label) – change

Figure 3. Explorer home page

The blocks here are called as streams. It shows the number of streams available such as public keys which will store the subscribed user's public key to be used for RSA encryption. It is possible to create many streams but deleting data in stream is not possible.



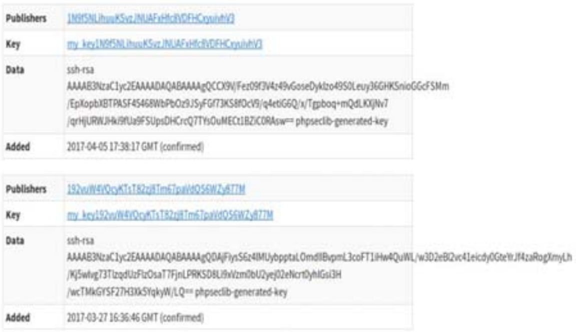
Subscribed streams

Name	root
Created by	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877M
Items	0
Publishers	0

Name	publickeys
Created by	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877M
Items	4
Publishers	4

Name	data
Created by	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877M
Items	160
Publishers	4

Fig 4. Available data streams

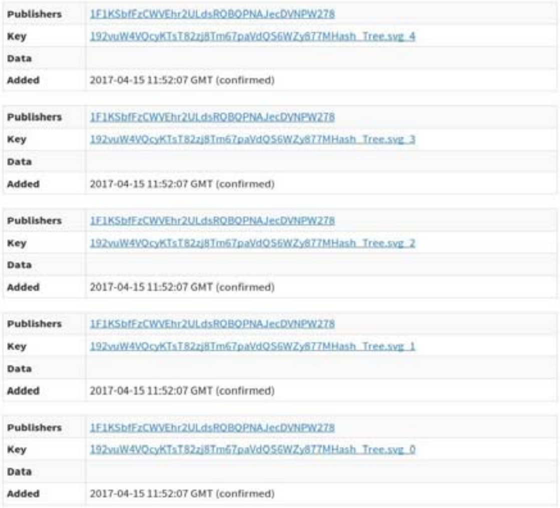


Publishers	1N9fSNLihuuKSvzJNUAFxHfc8VDFHCxyuivhV3
Key	my_key1N9fSNLihuuKSvzJNUAFxHfc8VDFHCxyuivhV3
Data	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQCCX9VjFez09f3V4z49vGoseDyklzo49S0Leuy36GHKSnioGGcFSMm /EpXopbXBTPASF45468WbPbOz9JSyFGf73KS8fOcV9/q4etiG6Q/x/Tgpboq+mQdLKXjNv7 /qrHjURWJHki9fUa9FSUpsDHCrcQ7TYsOuMECt1BZiC0RAsw== phpseclib-generated-key
Added	2017-04-05 17:38:17 GMT (confirmed)

Publishers	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877M
Key	my_key192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877M
Data	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDAjFiysS6z4lMUybpptaLOmdlIBvpmL3coFT1iHw4QuWL/w3D2eBl2vc41eicdy0GteYrJf4zaRogXmyLh /Kj5wlvg73TlzqdUzFlzOsaT7FjnLPRKSD8Li9xVzm0bU2yej02eNcrt0yhlGsi3H /wcTMkGYSF27H3XkSYqkyW/LQ== phpseclib-generated-key
Added	2017-03-27 16:36:46 GMT (confirmed)

Fig 5. Public key streams

It stores the public key of all subscribed users. Any new user must subscribe into this stream and must share their public key.



Publishers	1F1KSbfFzCWVEhr2ULdsRQBQPNAJecDVNPW278
Key	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877MHash_Tree.svg_4
Data	
Added	2017-04-15 11:52:07 GMT (confirmed)

Publishers	1F1KSbfFzCWVEhr2ULdsRQBQPNAJecDVNPW278
Key	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877MHash_Tree.svg_3
Data	
Added	2017-04-15 11:52:07 GMT (confirmed)

Publishers	1F1KSbfFzCWVEhr2ULdsRQBQPNAJecDVNPW278
Key	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877MHash_Tree.svg_2
Data	
Added	2017-04-15 11:52:07 GMT (confirmed)

Publishers	1F1KSbfFzCWVEhr2ULdsRQBQPNAJecDVNPW278
Key	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877MHash_Tree.svg_1
Data	
Added	2017-04-15 11:52:07 GMT (confirmed)

Publishers	1F1KSbfFzCWVEhr2ULdsRQBQPNAJecDVNPW278
Key	192vuW4VQcyKTsT82zj8Tm67paVdQS6WZy877MHash_Tree.svg_0
Data	
Added	2017-04-15 11:52:07 GMT (confirmed)

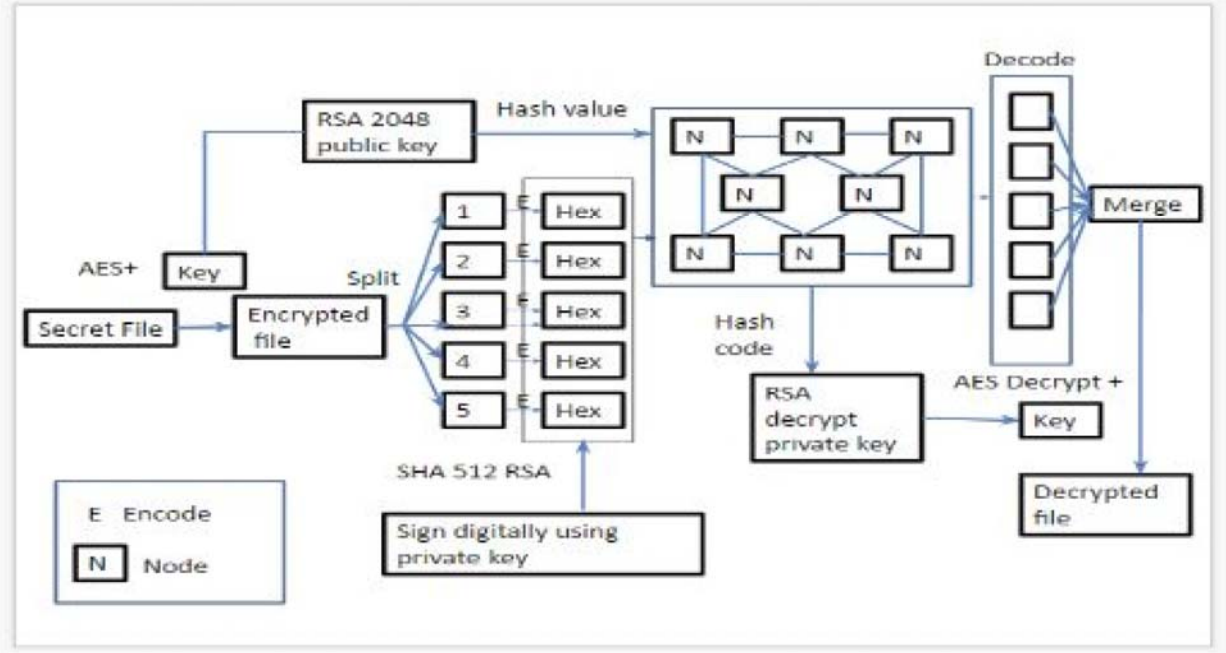
Fig 6. Data streams

Here the split files are named from 0 to 4. Also the fourth row implies that the transaction is confirmed. All the parts of the file must be confirmed and only then the transaction will be added to the blocks and the receiver can receive the file.

3.3 Implementation Architecture

In this concept a sender cannot directly send a file to the receiver. Whenever the user wants to receive a file, he can explicitly send a request to the sender. With the request, the receiver also shares his burn address along with its public key to the sender. Burn address is essentially the address to which the file has to be sent which will be generated by multichain framework on all nodes.

While the receiver requests the sender for the file, he can share his burn address along with it through by any means such as email or Facebook etc



.

3.3.1 Modular Design

Modular design defines the structure of the overall module. The overall system consists of the following modules:

1. Encryption/Decryption
2. Split/Merge
3. Hex Encode/Decode

3.3.1 Encryption / Decryption

At the sender's end, the input file is encrypted with symmetric cryptographic mechanism called AES+KEY, the keys are randomly generated then the randomly generated keys are encrypted with the Asymmetric cryptographic mechanism RSA(2048).

3.3.2 File Split/ Merge

The AES encrypted file is split into five equal halves. At the receiver each part is decrypted and then the decrypted parts are merged together.

3.3.3 Hex Encoding/ Decoding

Now the split parts are encoded with Hex encoding and then encoded values of five parts along with the RSA encrypted key will get stored on each blocks, created by miners.

Step 1: Initially the receiver has to request for the file by sharing his burn address.

Step 2: The sender will upload the file with the receiver’s address.

Step 3: The file uploaded goes through various modules.

Step 4: At the receivers side the file appears and the receiver can download it from his stream.

3.4 Security levels

Various levels of security are provided:

Level 0: Encryption of AES key with RSA allowing file access only by the receiver. Even if the files are available at all the blocks only the receiver can access the file. That’s beauty of the blockchain.

Level 1: Equal sized portions of the file are split and encoded with Hex encoding which is will be efficient way of sending files in the streams. Streams are nothing but blocks here.

Level 2: This is the most important level of security and the highest that can ever be obtained. Block chain network provides the highest form of security by making sure of the fact that when a file transfer takes place, the same has to be confirmed by all the nodes present in the network. All the nodes can surely see that a transaction is taking place between the sender and the receiver. But they can definitely not interfere or tamper the file in anyway. No node can snoop into the file and see what is being sent. This level of security provides a guarantee that only legitimate files can be transferred through the network. This legitimacy can be confirmed through the process of mining. Generally in a public bitcoin network, miners require the capability to perform highly intensive and sophisticated mathematical operations. In this case, since it is a private blockchain mining is not that complex and hard. Also the more number of miner the faster the file transfer will occur. In a file sharing application, miners have to determine who are the actual senders of files and then confirm the transaction that is taking place. We have constructed a private block chain network with few nodes at different places of the world and used a mechanism of mining which takes in any node which joins a block chain network to be a miner. Like a bitcoin explorer we also have an explorer to record all the transactions made. Each transaction contains values such as size, received time, mined time and whether or not the transaction is included in a block. By default, the transaction is unconfirmed. In this case the file is split into five equal part so all the five parts has to be confirmed by the miners only then the receiver can receive the file.

Level 4: At the receiver's side, to avoid leakage, strong algorithms from cryptography are implemented. Since the file is encrypted and split, the hackers or intruders cannot get access to the original file. It provides maximum integrity as well as security. Even if the file is tampered anywhere in between any nodes the root hash value will change using the Merkle tree data structure. The root hash will not match with the original hash and therefore the transaction does not get confirmed and remains tamper proof.

Thus all these levels of security will provide the most secure form of a file transfer. Even after the implementation of various algorithms that span a diverse and wide area of cryptography as well as Blockchain technology, it is still faster and secure than SFTP and TOR. In spite of the sophisticated methodology involved, the process clearly worked out to be much quicker than the normal file sharing applications which are less secure. From the time taken for file transfer, it is even possible to expect blockchain to be an essential technology in shaping tomorrow's applications as long as security is perceived to be the primary concern in the world.

4. Proof Of Concept(POC)

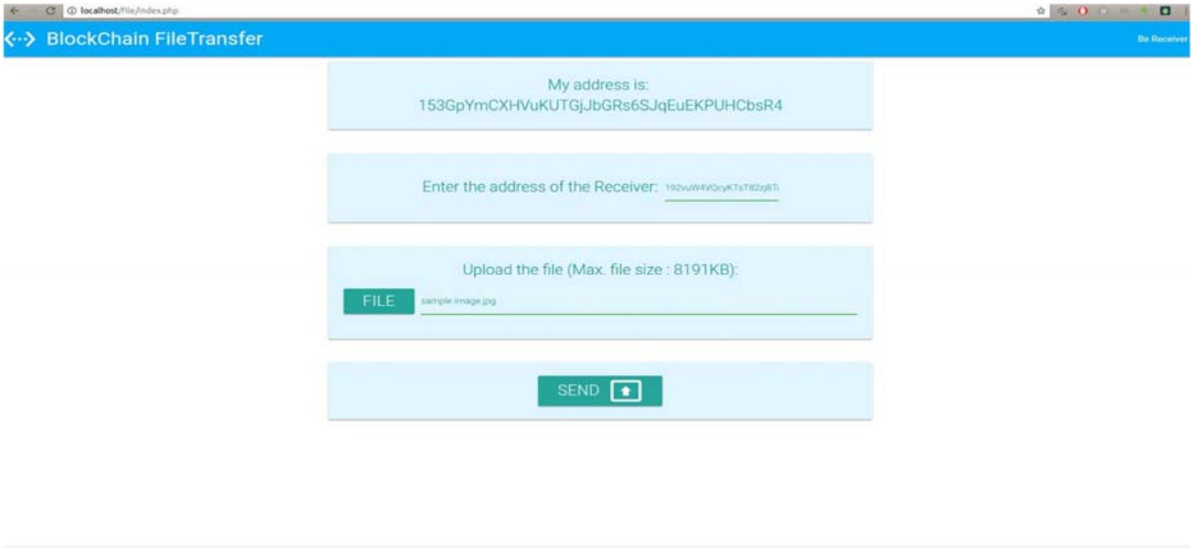
The security proof of Concept can be measured using a popular packet sniffing tool called wireshark[17]. When tried with normal FTP file transfer and SFTP the blockchain file transfer is much secured. In FTP the username and passwords are sent in plain text and can be easily seen.

4.1. Wireshark Sniffing

Wireshark[17] is an open source and very powerful packet analyser.

While analyzing the packet it is seen that the blockchain file transfer doesn't simply use FTP protocol. Also the files from the stream can be downloaded using Curl request in Java/PHP. Curl is faster and secure than FTP. Also it is secure and faster than SFTP.

Consider a scenario where a rogue node or hacker tries to pretend as a legitimate node in a private blockchain and tries to sniff traffic and decodes the data passed between the nodes. As a result a hacker cannot decode the packet and it seems unbreakable.

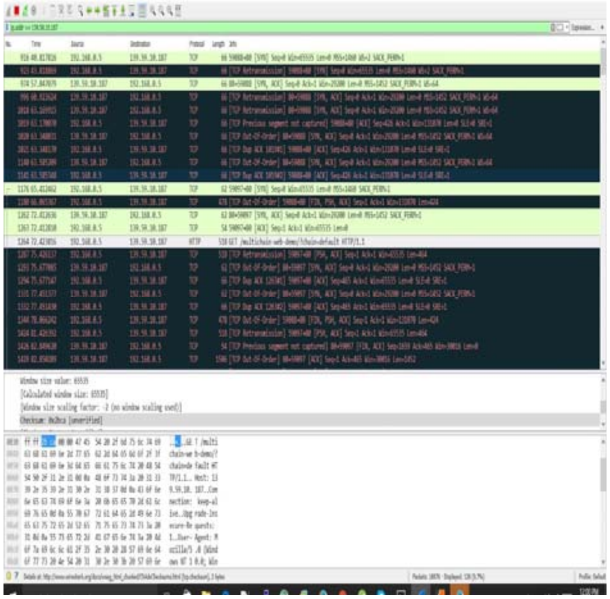


My address is:
153GpYmCXHVuKUTGjJbGRs6SJqEuEKPUHCbsR4

Enter the address of the Receiver: 192vuW4VQcyKTsT82zj8Ti

Upload the file (Max. file size : 8191KB):

sample image.jpg



5. Speed Comparison

The speed comparison is tested with the input file size of 250mb pdf file at the internet speed rate of 30 mbps. In the blockchain test environment it has 4 nodes and all the nodes were given miners permission.

Table 5.1. Speed comparison

FTP	STFP	Blockchain
2min 30 seconds	**3 min 10 seconds**	**2mins 5 seconds**

It is important that the more the number of miners, faster is the file transfer. The receiver can receive the file only if the transaction is confirmed by the miners.

6. Future Works

Since this concept is open source, we have implemented it using Php and Java. It can also be integrated into an android environment and developed into an Android app. The only problem with deployment in Android smartphones is memory handling. Mobile does not have a large memory to create and handle ‘n’ number of blocks. It is very challenging and and if it is implemented in the future, it will create a new revolution in android. Also this Blockchain concept can be integrated into various banking transactions to make them more secure and anonymous. Since the file is split and encoded with hex encoding, this could also create a possibility for compressing huge volumes of data into smaller sizes to further enhance the performance of file transfer in future.

7. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

8. REFERENCES

[1] KONSTANTINOS CHRISTIDIS AND MICHAEL DEVETSIKIOTIS (2016). BLOCKCHAINS AND SMART CONTRACTS FOR THE INTERNET OF THINGS[Online]. Available:http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7467408

[2] MultiChain Private Block chain-White Paper[Online]. Available:http://www.multichain.com/white-paper .

[3]. Svein Ølnes and Vestlandsforsking(2016). BEYOND BITCOIN[ISSN: 1894-7719 Online]. Available:http://ojs.bibsys.no/index.php/Nokobit/article/view/264/228

[4]. Understanding Public Key Cryptography [Online]. 2005 Available:https://technet.microsoft.com/en-us/library/aa998077(v=exchg.65) .aspx

[5]P.Franco (2014) Understanding Bitcoin: Cryptography, Engineering and Economics [Published] .Available: New York, NY, USA: Wiley

[6]. Incentives build robustness in bittorrent[Online]. Available:http://www.bittorrent.org/bittorrentecon.pdf

[7].P. Maymounkov and D. Mazieres(2002). Kademlia: A peer-to-peer information system[Online]. Available:https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-ln cs.pdf

[8].R.C. Merkle (1980). Protocols for public key cryptosystems[Online]. Available:http://www.merkle.com/papers/Protocols.pdf

[9].Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park, Kari Smolander(2016).

Where Is Current Research on Blockchain Technology?—A Systematic Review[Online].

Available:http://dx.doi.org/10.1371/journal.pone.0163477

[10].Shawn Wilkinson, Tome Boshevski, Josh Brandof, James Prestwich, Gordon Hall, Patrick Gerbes, Philip Hutchins, Chris Pollard(2016).

[11].STORJ: A PEER-TO-PEER CLOUD STORAGE NETWORK [Online].

Available: https://storj.io/storj.pdf

[12]. B. Cohen(2003).

Incentives build robustness in bittorrent[Online].

Available:http://www.bittorrent.org/bittorrentecon.pdf

[13]. Wood, D.G.(2014).Ethereum: A Secure Decentralised Generalised Transaction Ledger[Online].

Available: http://gavwood.com/paper.pdf

[14]. M. Swan(2014).

Machine ethics interfaces: An ethics of perception of nanocognition[Online].

Available:http://ieet.org/index.php/IEET/more/swan20151101

[15].Euro Banking Association report(2015). Cryptotechnologies, a major IT innovation and catalyst for change[Online].

Available:https://www.abeeba.eu/downloads/knowledge-and-research/EBA_20150511_EBA_Cryptotechnologies_a_major_IT_innovatio n_v1_0.pdf

[16]. Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timón, J., Wuille, P(2014).

Enabling blockchain innovations with pegged sidechains[Online]. Available: https://blockstream.com/sidechains.pdf

[17].WIRESHARK https://www.wireshark.org/download/docs/user-guide-a4.pdf