Impact Factor (SJIF): 5.301

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444
Volume 6, Issue 4, April-2019

Secure Multi Level Encryption with Blockchain in Distributed system

Ranjan P
Department of Information Technology
Pondicherry Engineering College
Puducherry, India
ranjanrab@gmail.com

Dr. M.S Anbarasi.
Associate professor
Department of Information Technology,
Pondicherry Engineering College
anbarasims@pec.edu

Prabhat kumar
Department of Information Technology
Pondicherry Engineering College
Puducherry, India
Pk4153639@gmail.com

Vijay T
Department of Information Technology
Pondicherry Engineering College
Puducherry, India
Vijaypondy952gmail.com

Abstract-- As we technologically progress, it is becoming extremely common for both businesses and individuals to use cloud storage services like Google Drive, Dropbox, iCloud and Amazon for storing their files. These services offer them the convenience of storing and retrieving their files from anywhere in the world using the Internet at relatively low costs, but they have introduced their own security and privacy issues because of the way these files storage systems are implemented.

In this paper, we draw inspiration from the *Blockchain* distributed ledger technology and various file distribution protocols to explore blockchain as an alternative strategy for file storage that offers security, privacy and reliability. For implementation, we look towards the powerful IPFS, offering node distribution. We design and implement a basic prototype of the suggested strategy, and discuss the results and the challenges of the approach.

Keyword- RSA; SHA256; IPFS; blockchain;

I. INTRODUCTION

Let's start with the Blockchain from where it all started. Going back to 90's the first work on implementing cryptographic secured chain of blocks was described in 1991 by Stuart Haber and W. Scott Stornetta [3]. After about a decade and a half the first distributed blockchain was theorized by an anonymous person or a group called as Satoshi Nakamoto in 2008 and during the following year implemented Bitcoin which then served as a public ledger for all transactions.

Blockchain: Blockchain is a relatively simple concept based on the Merkle Tree data structure, but when implemented it leads to a whole new set of technologies. The technology is still in its early stages and there are various ways to implement it which completely depend upon the problem statement. "*At the superficial level blockchain allows a network of computers to agree at regular intervals on the true state of distributed ledger*,"[5]. What exactly is a distributed ledger? It is a type of database which is shared, replicated, and synchronized among members of a network. Whereas every record in a distributed ledger has a unique timestamp and cryptographic signature attached to it which identifies it uniquely [8]. Such collection or block can contain different types of data varying from account credentials, transactional records or other arbitrary data. The security of this block is maintained through various cryptographic algorithms and game theory techniques [7]

RSA: RSA today is used in hundreds of software products and possibly used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a flexible size encryption block and a flexible size key[2]. The key pair is acquired from a very large number, n, that is the product of two prime numbers chosen in accordance to special rules. Since it was introduced in 1977, RSA has been broadly used for establishing secure communication channels and for authentication the identity of service provider over defenseless communication medium. In the authentication scheme, the server implements public key authentication by taking signature of client on a unique message from the client with its private key, thus creating what is called a digital signature.

Cryptography:

Cryptography may be defined as a study of methods or techniques that involve security of data to be transmitted across a network. Cryptography involves encoding and decoding of data to prevent it from any alteration, modification or just listening of data by a third party. Cryptography is one of the main techniques that is being used in computer security that converts information from its normal form into an unreadable form.

Encryption:

Encryption is the technique that converts any readable format into non-readable format. It takes any plain text and converts it into non-readable format with the use of any algorithm which may or may not use any key (or keys).

Decryption:

Decryption is just the inverse process of encryption. It converts non-readable format into readable form by taking the encrypted text known as cipher text as input and applying the decryption algorithm to it, giving us the original plain text.

II. LITERATURE SURVEY

2.1 Survey on multi-level security using RSA and SHA256:

In Cloud Storage any organization's or individual's data is stored in and accessible from multiple distributed and connected resources that comprise a cloud. To provide secure communication over distributed and connected resources, encryption algorithm plays a vital role. It is the fundamental tool for protecting the data. Encryption algorithm converts the data into scrambled form by using "the key" and only user have the key to decrypt the data. In Symmetric key encryption, only one key is used to encrypt and decrypt the data. Another technique is known as asymmetric key encryption where two keys- private and public keys are used. Public key is used for encryption and private key is used for decryption.

2.2 Survey on IPFS Distributed files storage:

For file storage and sharing between nodes, we look towards IPFS. IPFS is a protocol which enables the reliable and quick transfer of data from many to many peers in distributed network. It is one of the most widely used peer-to-peer program ever made. The man behind IPFS thought the idea of breaking down files in chunks, encrypting every chunk and storing them in different locations can be used in file sharing.

2.3 Survey on file storage on Blockchain:

There are two main ways to store a document on the blockchain. One option is to store the entire document itself on-chain. Alternatively, you can store a hash of it on the blockchain

The problem with storing whole documents on a blockchain is because of something called *access latency*. Fully decentralized public blockchains have thousands of nodes. Unfortunately, the benefits that come with this number of nodes also results in a corresponding increase in latency. Any file storage, including documents, needs to have low latency otherwise the system becomes clogged up, slow, and expensive to use.

The most efficient method is to store a document's hash on-chain while keeping the whole document elsewhere. The document could be stored on a distributed file storage system. You would put the document through a secure hash algorithm like SHA-256 and then store the hash in a block. This way you save a huge amount of space and cost. Additionally, if someone tampers with the original document.

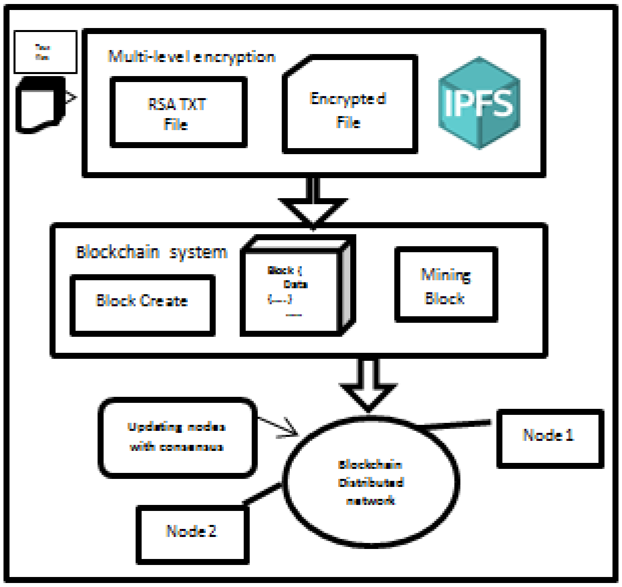
The change in input would result in a completely new hash value; different from your original document.Hash values are far smaller than whole documents and so is a vastly more efficient blockchain storage method. It also scales efficiently. For storing multiple documents, you can put the hashes into a distributed hash table, which you then store on-chain.

3.1 Problem definition

Implementing a Distributed Blockchain File Storage (DBFS) system which aims on targeting three main concerns of the current world which are *security, privacy and reliability*. The application is designed as a private blockchain, allowing it to be much more secure and efficient compared to other types, and putting the reliability of the data in to the hands of the organization running it

3.2 System model

The system will consist of a basic user interface for the end users to interact with the files they will be uploading to the server. The system consists of 2 main components, a client application that performs encryption and signing of data; and a backend node server where all signatures and blocks are verified and validated, and the consensus between all nodes in a network is maintained. Our custom Blockchain implementation would allow us to store data references, its timestamps, and cryptographic signatures of the users performing those actions, using a hybrid of both Symmetric and Asymmetric key cryptography. This also provides a powerful way to maintain and check the data consistency and keep track of its changes, making it impossible to alter the data history and its integrity. For efficient delivery of data, we'll use IPFS to transfer the data from many to many peers in a distributed network.



3.2.2 Architecture diagram for Secure Multi Level Encryption with Blockchain in Distributed system

3.3 File encryption with asymmetric algorithm and SHA256

The encryption module provides an interface to work with the RSA Asymmetric cryptography and SHA256 to secure the file with public and private key. These keys consist of an ordered pair of integers (a, b). For encrypting, let's call these numbers "(e, n)" and for decrypting, "(d, n)".

3.4 The *Block and Crypto* system

The *Block* module defines the block struct, which contains the type of block, creation timestamp, stored data, owner's encoded public key, hash of the previous block, cryptographic signature verifying authenticity of the data and the hash of the entire block.

The cryptographic actions like signing, hashing and verifying blocks are performed using the *Crypto* module which implements Asymmetric key cryptography and the SHA256 hashing function using the open- source *RsaEx* and *ExCrypto* libraries, The *Blockchain* module provides an interface to work with the Blockchain data structure itself consisting of Linked List of many *Blocks*. The blockchain starts with a zero block, a special data structure to seed the rest of the chain

3.5 The *Proof of work and RAFT consensus*

It context consists of a few modules responsible for ensuring that all nodes in the network are in sync. The *Proof of Work and raft consensus* module implements the actual function calls responsible for sending and retrieving blockchain blocks and their associated files between nodes and updating their internal states.

3.6 Input & Output

Input: The proper formatted text files. The user will choose the file to be uploaded

Output: Private Blockchain which stores the attributes of the text file

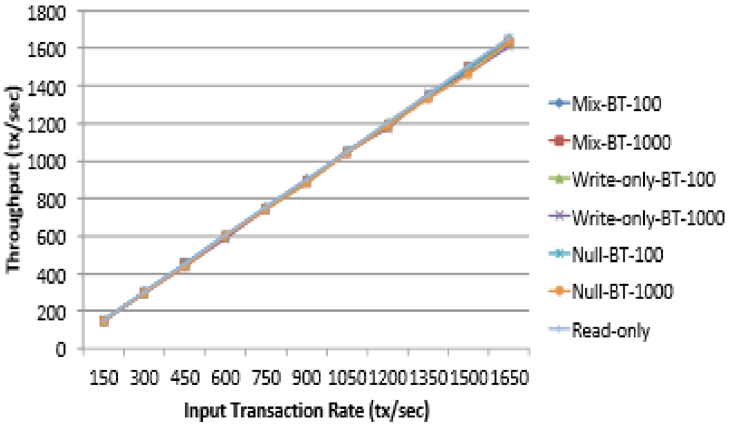
IV EXPERIMENTAL RESULTS AND EVALUATION

4.1 Simulation of environment

Implementation of our proposed system the carried out using NETBEANS (java) and ANACONDA (python) running on a personal computer with a 2.07 GHz Intel ® Core ™, i3 CPU, 4GB ram and windows 10 as the operation system. The metric calculation determines the quality of the proposed security model

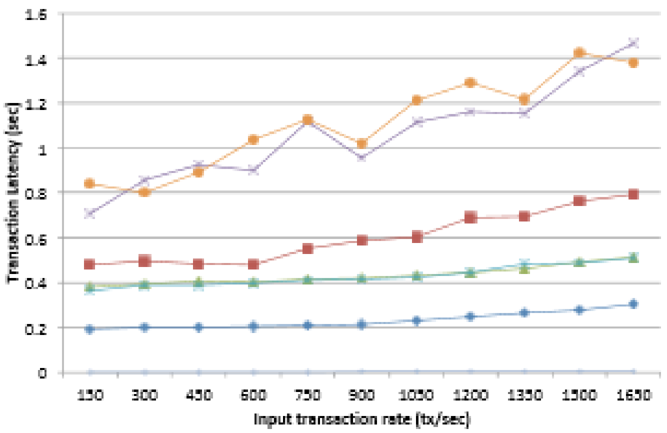
4.2 Latency and Throughput with RAFT Consensus

It is seen from the results that IBFT actually provide slightly greater throughput until an input is 1500 transaction per second, which is contrary to the expectation. RAFT performs little better when the load is increased beyond 1650 transaction per second. Both algorithms scale well with RAFT gives little advantage for very high input transaction rates. The transaction latencies however are significantly higher for IBFT consensus. For most data points IBFT latencies are almost twice or more than twice inspite of both RAFT and IBFT having the similar block time setting of 1 second.



(a) Transaction throughput

Fig 4.2.1 Throughput measurements with RAFT

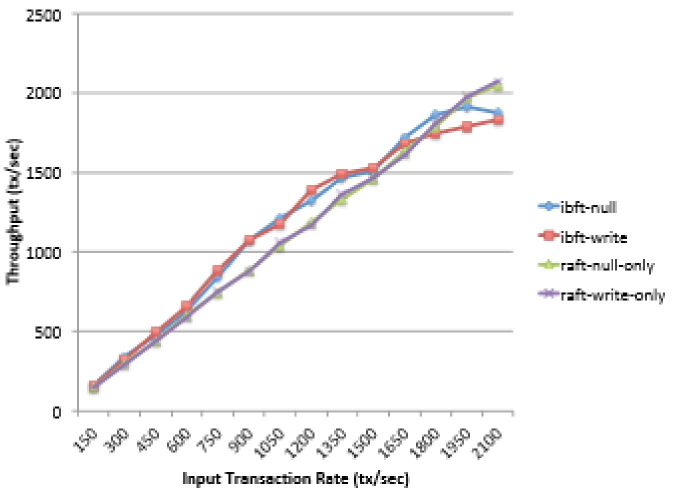


(b) Transaction latency

Fig 4.2.1 Throughput measurements with RAFT

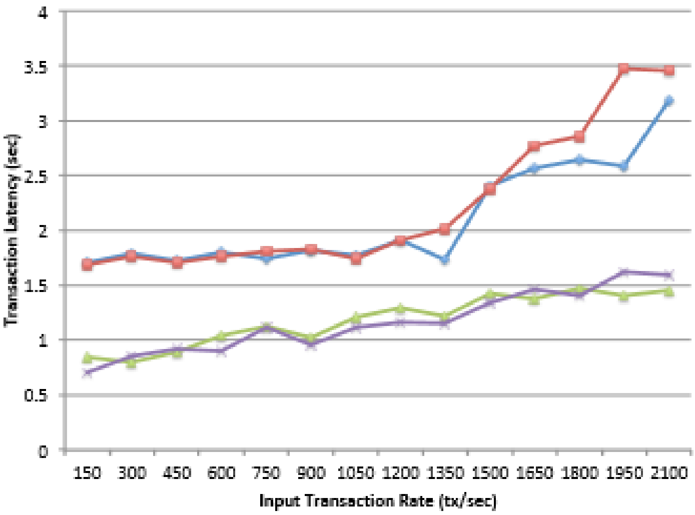
4.3 RAFT versus IBFT

It is seen from the results that IBFT actually provides slightly greater throughput until an input of 1500 transaction per second, which is compliments the expectation. RAFT starts to perform slightly better when the load is increased beyond 1650 transaction per second. Both algorithms scale well with RAFT produce a little more advantage for very high input transaction rates. The transaction latencies however are significantly higher for IBFT consensus. For most data points IBFT latencies are almost twice or more than twice inspite of both RAFT and IBFT produce the similar block time setting of 1 sec.



(a) Transaction throughput

Fig 4.2.1 Throughput measurements of RAFT versus IBFT consensus



(b) Transaction latency

Fig 4.2.1 latency measurements of RAFT versus IBFT consensus

V. CONCLUSION AND FUTURE WORK

Secure Multi Level Encryption with Blockchain in Distributed system was certainly an exciting project to work on, but there's only so much that can be accomplished by an individual in the span of 4 months. The implementation is a very naïve version of the blockchain by drawing inspiration from other extremely popular projects and protocols, and though there were a lot of issues in implementation initially, the final application still turned out to be a great success.

There is still room for a lot of improvements, but we leave that for the future work on this topic.

There's opportunity for a lot more work and improvements on this project. Some of the other features that can be added are:

- More types of blocks – we can allow creation of a diverse set of blocks that could allow deletion of previously shared files, publicly accessible sharing and multi-key user-based sharing of files.
- Advanced implementation of file distribution across nodes – to allow faster and more efficient transfer of files by breaking them into smaller chunks (like the original BitTorrent protocol).
- Improved Conflict Resolution – to resolve conflicts and differences in "Blockchain Branching" when the blockchain data on one node completely diverges from another.
- Better Key Handling – By using compression, caching, removal of unnecessary data, and support for encrypted ECDSA Keys .

Clearly, there's much more possible in the world of Blockchains and File Storage

REFERENCES

[1] Pradip Kumar Sharma1, Mu-Yen Chen2, And Jong Hyuk Park "*A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT*", Feb 1, 2018, VOLUME 6 , p115-124

[2] Kalyani P. Karule, Neha V. Nagrale "*Comparative Analysis of Encryption Algorithms for Various Types of Data Files for Data Security*", International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-2, February 2016 ISSN: 2395-3470

[3] Blockchain Whitepaper https://www.blockchain.com/whitepaper/inde x.html

[4] Nishanth.G.Hhulwan, Prof.Sachin Chavan "*Blockchain-Based Security Architecture for Distributed Cloud Storage*", IEEE International Symposium on Parallel and Distributed Processing with Applications, 2017

[5] Ali Dorri, Marco Steger, Salil S. Kanhere, and Raja Jurdak, "*BlockChain*: A Distributed Solution to Automotive Security and Privacy", IEEE Communications Magazine, 2018

[6] Pritam Singh Negi,"*A survey on Data Storage and Retrieval in Cloud Computing*", International Journal on Computer Science and Engineering, ISSN: 0975-3397 Vol.8 No.7 Jul 2016

[7] Kalyani P. Karule1, Neha V. Nagrale2,"*Comparative Analysis of Encryption Algorithms for Various Types of Data Files for Data Security* "[7] MIT Sloan (2017) – Blockchain explained. MIT Sloan Assistant professor Christian Catalini http://mitsloan.mit.edu/newsroom/articles/blo ckchain-explained/

[8] IBM developerWorks (2016) – Blockchain basics: Introduction to distributed ledgers by Sloane Brakeville and Bhargav Perepa https://www.ibm.com/developerworks/cloud/li brary/cl-blockchain-basics-intro-bluemixtrs/index.html