Using blockchain technology for file synchronization

Khan, Md Ibrahim; Faisal, Fahad; Azam, Sami; Karim, Asif; Shanmugam, Bharanidharan; De Boer, Friso

Published in:
IOP Conference Series: Materials Science and Engineering

DOI:
10.1088/1757-899X/561/1/012117

Published: 12/11/2019

Document Version
Publisher's PDF, also known as Version of record

Using blockchain technology for file synchronization

Using blockchain technology for file synchronization

MD. Ibrahim Khan[1], Fahad Faisal[2], Sami Azam[3], Asif Karim[4], Bharanidharan Shanmugam[5], Friso De Boer[6]

[1&2]Department of Computer Science and Engineering, Daffodil International University, 4/2 Sobhanbag, Dhaka, Bangladesh
[3,4,5,6]College of Engineering, IT and Environment, Charles Darwin University, Ellengowan Drive, Casuarina, NT, Australia

E-mail: ibrahim15-4739@diu.edu.bd, fahad.cse@diu.edu.bd, sami.azam@cdu.edu.au, asif.karim@cdu.edu.au, bharanidharan.shanmugam@cdu.edu.au, friso.deboer@cdu.edu.au

Abstract. Modern storage technology has shifted from traditional offline state to cloud based technology since some time now. Because of this transition, the present society is now more dependent on the online storage solutions. Synchronization of files and keeping a history of changes are critical parts of any cloud system. Therefore, an implementation of Blockchain Technology with traditional file synchronization and versioning system can be extremely fruitful. Blockchain is not a new technology, but recently its importance has sky-rocketed as the society is moving towards the decentralized World Wide Web. Blockchain is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way" [1]. Blockchain provides immutable data storage and access with the combination of Proof-of-Work [2, 3]. Due to such appealing features, the study undertaken here investigates and proposes a Blockchain based resilient cloud storage solution that makes a sound utilization of various properties fundamental to any Blockchain based framework.

1. Introduction

The adoption of Cloud storage is soaring and this modern era is a milestone for uninterrupted online connectivity. But the online connectivity is not always accessible or fast enough comparing to the offline physical storages. There are issues of data speed and data limit to wireless data usage. That is why it is important to implement a synchronization system that involves minimal data transfer for CRUD operations to the cloud server. There are multiple software available for data and file synchronization, such as **rsync**[4] for Linux. The proposed system is quite similar to **Differential Backup** [5], where the last data entry is the changes made to the previous data. In the proposed system, Blockchain Technology will be employed to store those differential data. Blockchain will act as an immutable data source, where only new blocks of data is added but never change any of the previous blocks. Each update will be added as a new block in the chain. The link from one block to another is similar to DAG [6], one block can point to the immediate previous block as well as any other block in the existing chain. By embedding version information within the block, one can keep track of the file version and also maintain a continuous chain of blocks. Thus, each version will create a virtual sub-chain within the main Blockchain. Every block will contain signature of the user that made the update to the Blockchain. This signature will ensure the identity of the user. The implementation of Proof-of-Work will be done by contributors and users. Each update to the block will be sent to other users. The calculation for Proofof-Work will be evenly distributed to other contributors and users. The devices that will be used to carry out these tasks are the nodes in the network. Commands will traverse freely over the network. Upon success, the node will reply with result to network. Thus a distributed network will be created where every contributors and users will hold the complete file stored in a Blockchain. Blockchain is the base for the most used cryptocurrency, Bitcoin [20].

2. Data block format

A block is the smallest chunk of data that will be added to the Blockchain. Information that is contained in a single block contains is demonstrated in Figure 1 and illustrated below:

- *Index:* Serial or index number of the current block
- *Data:* Binary data that will be included in the current block
- *Data Length:* Length of the included data
- *Hash:* Generated hash of the current data along with data from the previous block
- *Index for previous block:* Serial of index number of previous block [19]
- *Nonce [7]:* Cryptographic nonce value generated from Proof-of-Work
- *User signature:* Digital signature of the user who made the changes
- *Version Information:* Version information for the changes

Block
Index
Data
Data Length
Hash
Index of previous block
Nonce
User Signature
Version information

Figure 1. Basic block information.

In client-side, all blocks from the Blockchain will be downloaded on the local storage. Then the user can construct the original file, or a specific version of the file from iterating the blocks in the Blockchain.When the user makes any changes to the file, a new block will be generated and an update will be pushed through the network. The differential data can be generated by using pre-existing tools or other suitable methods. For example, "diff" [12], can be used, which is a well-known utility tool that displays difference between two files.

3. Synchronization between server and nodes

The proposed network is a hybrid of decentralized and distributed network. A centralized server will store a copy of the main Blockchain, along with the connection information, P2P [8] communication information of other connected users and contributors. Users and contributors can directly communicate with the server to discover other nodes in the network, or they can use previously discovered nodes for P2P communication system to communicate to each other. Once an update is made, all nodes in the P2P communication will contribute to the Proof-of-Work and relay the result back to the other nodes in network for final update to the Blockchain.

In Figure 2, it can be seen that the server acts as the primary medium for discovery of the other nodes. But users and contributors can communicate with each other in P2P to maintain the Blockchain network.

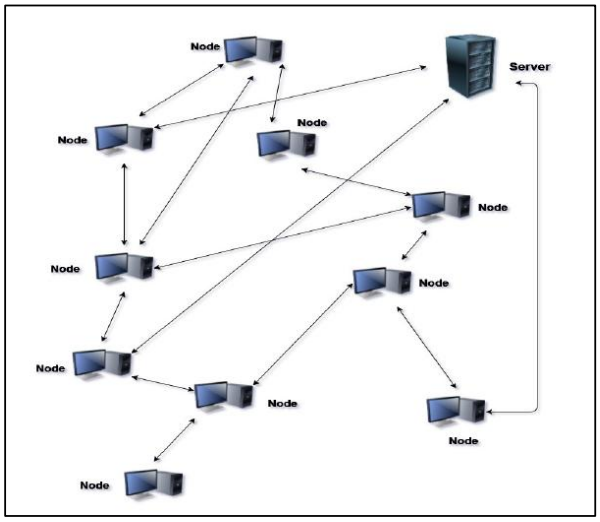

Figure 2. Decentralized and Distributed network

4. Version Control

Version control is implemented by embedding version information within the block that contains the differential data.

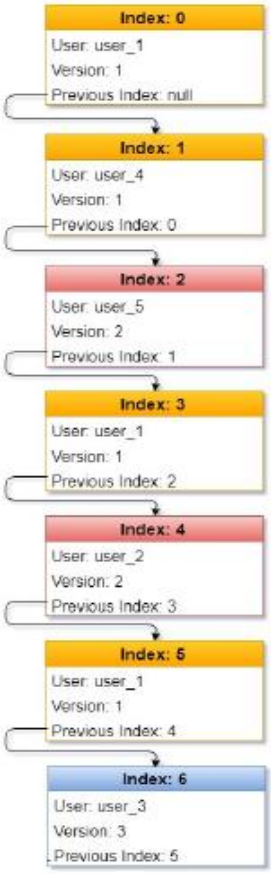

Figure 3. Version control demonstration

It can be thought of as a virtual sub-chain is created inside the original Blockchain for each version. Although every update made to the file by each user and contributor is stored in one Blockchain, but one can access and maintain different versions of the file by simply scanning through comparing the version data in the Blockchain. In Figure 3, an example Blockchain is shown with corresponding data values. Similar colour in Figure 3 represents the same block for similar version of the file.

5. Multiuser access and user authentication

Every user can add new updates to the Blockchain network. Each update block requires the Digital signature [9] of the user or contributor who is making the update. In Figure 3, it can be observed that every block has user information that stores the signature of the user. Boxes with similar colour in the figure contain data of the similar version of the file. For example, DSA (Digital Signature Algorithm) [10] can be utilized to generate public and private key for the user. In the server, private keys for all users and contributors will be stored and shared with each other. While adding new update, signed data with the public key of the user will be stored in the updated data block. Anyone can then verify the user by using the private key that was shared earlier.

6. Implementation of consensus

A threat modelling will aid in identifying different assets, users and gateways to access the system, thus giving us an overall view of various potential threats faced by the system [5]. The threat modelling that is followed by the process is as follows [12]:

- To Identify the Smart Parking System Assets
- Decompose the IoT System
- Identify Threats

The most popular implementation of consensus system is the Proof-of-Work. Although there are many consensus systems still present, but Proof-of-Work is exclusively used in the open Blockchain network for verification of a legitimate block. Although it is possible to implement other consensus systems such as PBFT (Practical Byzantine Fault Tolerance) and FBA (Federated Byzantine Agreement), but only Proof-of-Work will be discussed. A proof-of-work (PoW) system is "an economic measure to deter denial of service attacks and other service abuses such as spam on a network by requiring some work from the service requester, usually meaning processing time by a computer" [2].

The PoW implementation in the proposed system is quite similar to HashCash [11].

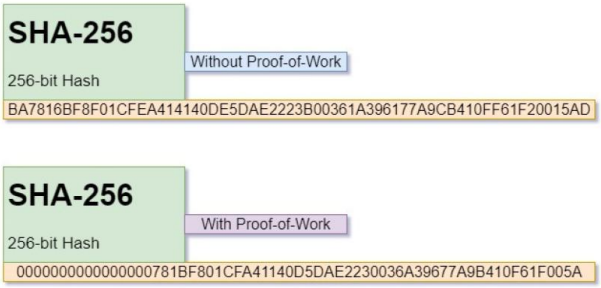

Figure 4. Hash with and without proof of work

The calculated hash of a data block will have a predefined prefix. For example, zero (0) or a series of predefined letters or numbers or both can be used. In Figure 4, it can be seen that a SHA-256 hash with Proof-of-Work implementation has 16 zeros as prefix. In the process of Proof-of-Work, the "Nonce" [7] is generated which represents the proof that some work has been done in calculation to obtain the hash result. The calculation for Proof-of-Work is distributed among the users and contributor in the network. So, the users of the network will verify for each other and contribute to the calculation power, thus reliving the server of heavy work load. For experiment, if the first 20 bits of

the 256 bit SHA-256 hash digest are used for the PoW system, out of 2256 possible hash values, there will be 2246 hash values that satisfies this criterion. Thus, 220 hash calculations will have to be done on average, to add a new block to the Blockchain. To calculate such hash collision, the total time needed is exponential with the number of zero bits. So, it will double the amount of time needed to compute a hash with each additional zero bits. Hash function was initially used around 1960 in computing systems, various flavours of this function have become quite popular these days in multitudes of applications [13].

7. Security

The main objective of the proposition is to implementation or create more secure environment. Blockchain at present is one of the most secure ways to maintain and distribute data among multiple untrusted sources [14]. Along with an encrypted network of SSL/TLS [15] and digital signing by the user, this system presents a secure way to synchronize and maintain file by multiple or unlimited amount of untrusted users. With the inclusion of Proof-of-Work, this system prevents brute force attacks, DdoS [16] and spamming. As the system is distributed and decentralized, the network will continue to work even if the communication server fails. In Blockchain network, the "51% attack" is one of the few attacks that can be carried out. To secure the network, Byzantine Fault Tolerance [18] along with the enforcement of the number of users that contribute to add an update to the original Blockchain can be incorporated, so that Blockchain based threats such as "51% attack" [17] becomes preventable.

8. Difference between existing cloud/server based systems and proposed implementation

Table 1 below shows some major differences between proposed Blockchain based system and other Cloud/Server based system for file synchronization.

Table 1. Difference between existing cloud\server based systems and proposed solution.

	Blockchain based system	Cloud / Server based system
File integrity	File integrity is maintained within the chain itself	File integrity is maintained with external operation
Flow of communication	Centralized server is only needed for communication between different networks. Partially operable with out a centralized server	Heavily depends on the centralized server for all operation
Data Storage	File data is stored in the contributing member hosts or users storage	File data is stored in the server
Nature of storage	Only a constant amount of storage is used to store the blocks	Storage depends on the features and implementation technology. May use extra space to store data integrity information
Validation	It is possible to validate data integrity while offline	Data integrity cannot be validated while offline
Fault tolerance	Operational even if server or multiple contributor hosts are offline	Operation is completely stopped if the server is offline
Security	Greater security for decentralized and distributed architecture	System security depends on the implementation of the server

9. Experimental result

The performance of the proposed solutions has been measured using different experimental setup as illustrated below:

9.1. Total size of data file

As data files are stored on the blocks, extra storage space is needed for each block to store additional data. This enable the opportunity to predict the actual data size, as blocks occupy a specific amount of storage. Figure 5 graphically highlights the issue.

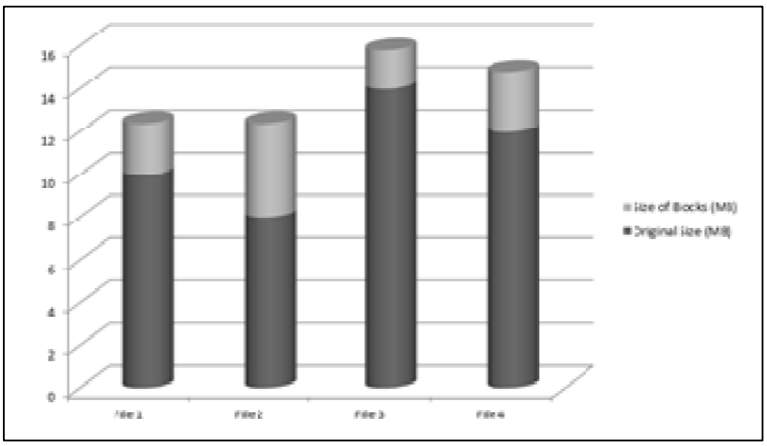

Figure 5. Blockchain file size

9.2. Data storage for multiple versions

Storing data chunks in the blocks enabled the work to save more space by removing duplicate data. In general versioning system, two versions of file occupy more than double the size of the original file.

In the proposed implementation, only the changes are stored in the Blockchain, thus only the needed amount of storage is required. In the experiment that has been conducted, it can be seen that, proposed implementation resulted in huge reduction of storage space. Figure 6 pictorially describes the comparison.

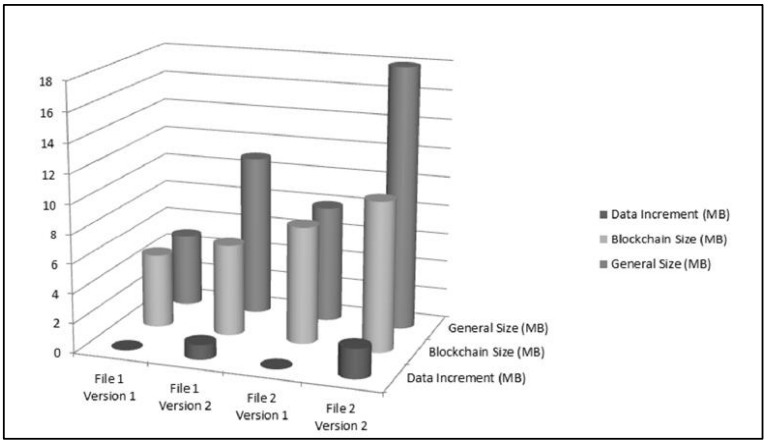

Figure 6. Storage occupation for multi version file

9.3. Overhead for block count

Storing minimal data in each block may increase the number of blocks to two or three times. As each block must acquire a specific amount of storage to store additional information, it might create computational and I/O overhead to the system. While experimenting, exponential increase of data size for using non-optimized and excessive block numbers has been observed. Figure 7 graphically shows the increase.

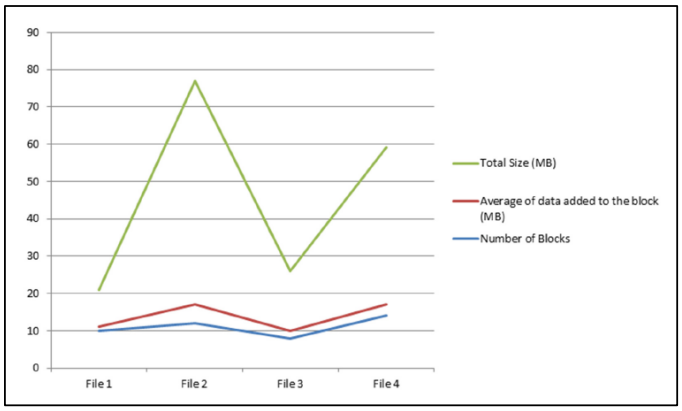

Figure 7. Overhead for excessive block number

9.4. Synchronization time

As data is stored differentially for appending and versioning, one can see a huge reduction of total synchronization time. After initial synchronization, only the incremental data and related additional block information are needed to be synched. Figure 8 shows total sync time for files.

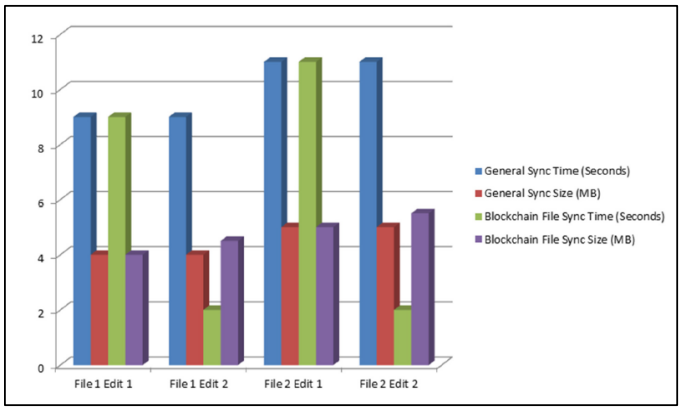

Figure 8. Total sync time for files

10. Conclusion

In this paper, a proposition has been made to implement Blockchain Technology with a decentralized and distributed network for file synchronization and version control. The implementation is strictly contained within the users and contributors of the system and requires minimal centralized server power. This implementation ensures contribution of calculative power and participation of users along with data immutability and security.

References

[1] Iansiti M and Lakhani K R 2017 The Truth About Blockchain Harvard Business Review, Harvard University, hbr.org/2017/01/the-truth-about-blockchain, accessed date: February 2, 2019.

[2] Coelho F 2007 An (Almost) Constant-Effort Solution-Verification Proof-of-Work Protocol based on Merkle Trees Progress in Cryptology – AFRICACRYPT 2008, p 80-93.

[3] Bentov I Lee C Mizrahi A and Rosenfeld M 2014 Proof of Activity: Extending Bitcoin's

Proof of Work via Proof of Stake [Ext. Abstract], ACM SIGMETRICS Performance Evaluation Review, 42(3).

[4] Rasch D Burns R 2003 In-place rsync: file synchronization for mobile and wireless devices, Proceedings of the annual conference on USENIX Annual Technical Conference, p 15-15.

[5] Nelson S 2011 Introduction to Backup and Recovery, Pro Data Backup and Recovery, p 1-16.

[6] Moffa G Catone G Kuipers J Kuipers E Freeman D Marwaha S and Bebbington P 2017 Using Directed Acyclic Graphs in Epidemiological Research in Psychosis: An Analysis of the Role of Bullying in Psychosis, Schizophrenia Bulletin, 43(6), p 1273-1279.

[7] Bellare M and Tackmann B 2016 Nonce-Based Cryptography: Retaining Security When Randomness Fails. Advances in Cryptology – EUROCRYPT, p 729-757.

[8] Han J 2014 Distributed hybrid P2P networking systems. Peer-to-Peer Networking and Applications, 8(4), p 555-556.

[9] Zhu L and Zhu L 2012 Electronic signature based on digital signature and digital watermarking, 5th International Congress on Image and Signal Processing.

[10] Ihwani M 2016 Information Security Model Based Digital Signature Algorithm with RSA Algorithm. Computer Engineering, Science and System Journal, 1(1), p 15-20.

[11] Back A 2002 HashCash - A Denial of Service Counter-Measure, Technical Report.

[12] MacKenzie D 2017 GNU diffutils - Comparing and Merging Files, https://www.gnu.org/software/diffutils/manual/, accessed date: February 3, 2019.

[13] Karim A 2017 Multi-layer Masking of Character Data with a Visual Image Key, International Journal of Computer Network and Information Security, 10(10), p 41-19.

[14] Stephen R and Alex A 2018 A Review on BlockChain Security. IOP Journal of Physics: Conference Series, 396, 012030.

[15] Dierks T Rescorla E 2008 The Transport Layer Security (TLS) Protocol, Version 1.2, RFC5246.

[16] Suriadi S Clark A Liu H Schmidt D Smith J and Stebila D 2011 Denial of Service Defence Appliance for Web Services. An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks, p 239-298.

[17] Ye C Li G Cai H Gu Y and Fukuda A 2018 *Analysis of Security in Blockchain: Case Study in 51%-Attack Detecting.* 5th International Conference on Dependable Systems and Their Applications (DSA).

[18] Chen L and Zhou W 2015 Byzantine Fault Tolerance with Window Mechanism for Replicated Services. 5th International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC).

[19] Zhai S Yang Y Li J Qiu C and Zhao J 2019 Research on the Application of Cryptography on the Blockchain. IOP Journal of Physics: Conference Series, 1168, 032077.

[20] Jose J Kannoorpatti K Shanmugam B Azam S and Yeo K C 2017 A critical review of Bitcoins usage by cybercriminals, IEEE International Conference on Computer Communication & Informatics.