Developing Blockchain Authentication for Hadoop using HDFS Client

Dr. Pramod Patil, Dr. Jyoti Rao, Mithun Kankal

Abstract: The Apache Hadoop information system uses Kerberos authentication protocol provided by MIT for authentication. There are various security issues in the Kerberos protocol which are remained unsolved like single purpose of failure, DDoS, replay attacks are some examples. It illustrate the potential security threats or vulnerabilities and huge information issues in victimization of Hadoop. Here authors meant to presents weakness of Kerberos implementations and identify authentication needs that may enhance the security of huge information in distributed environments. The developing mechanism will be a new perspective of using blockchain in Hadoop for authentication instead of Kerberos. The mechanism is relies on the rising technology of blockchain that overcomes the shortcomings of Kerberos. The author has utilized blockchain basic concepts and created HDFS client model of blockchain-based authentication mechanism for big data framework which can coexist along with Hadoop setup, where it describes and implemented private blockchain methodology which could be implemented for a private organization in there Hadoop setup. Also, it provides the various basic operational feature for blockchain admin and HDFS client for end-user along with distributed local authentication mechanism using blockchain.

Keywords: Big Data, Distributed Authentication, Hadoop, Security, Blockchain, Decentralized Authentication, Private Blockchain, Blockchain admin.

I. Introduction

Security of Big Data is significant on account of ceaselessly expanding trade of delicate information. Information is being gathered from various autonomous sources, where they are frequently than intertwined and broke down to produce knowledge. Henceforth, these information are a profitable resource in the present economy. Concerns have concentrated on security and assurance of delicate data, where these identify with new dangers to data security and embracing existing conventional safety efforts isn't satisfactory. The present authentication arrangement of Apache Hadoop uncovered the whole Big Data answer for a security issue because of Kerberos' framework vulnerabilities. Impediments of Kerberos are obvious in rendition 4 and early drafts of adaptation 5; Single point of failuare, replay assaults, key exposure, no secret key arrangement for Kerberos verification and time synchronization are vulnerabilities recognized. It present the auxiliary downsides and recognize verification prerequisites, for example, Blockchain that can improve security of Big Data in circulated situations.

The intent is to take advantage of blockchain technology which is decentralized or distributed in nature. Also it uses password-less/keyless authentication, data encryption in form of hash and does not need any third party or a central database. Blockchain authentication is introduced in number of different sector but it was first used for transaction verification in Bitcoin. Here Authors focus on the usage of blockchain as authentication providers and describes how the blockchain basic concepts could be used for developing model for distributed authentication mechanism for Hadoop, which is integrated with HDFS Client. The use of blockchain technology for authentication provider is at a very early stage right now, but it is increasing at a rapid pace. Blockchain uses the key-pair for user registration of identity. The User information is stored in hashes form and used for storing name or any other personal information. After that, whenever user tries to access system, the user information is verifies against blockchain hashes and checked whether provided information is true.

II. Literature Review

A. *Kerberos and Hadoop Cluster*

Hadoop Cluster are normally any set of tightly connected or loosely connected computers that work together as a single system is called Hadoop Cluster. In simple words, a cluster of computer used to deploy Hadoop is called Hadoop Cluster. Hadoop cluster is a computational cluster build for storing and analyzing huge amount of structure and unstructured and semi-structure data in a distributed computing environment. These clusters run on commodity computers which can available at low cost.

Hadoop works with a group of computers and each individual computer executes an independent operating system. Authentication for individual computer works with OS boundary. However Hadoop Cluster works across those boundaries. So, Hadoop should have separate a network-based authentication system. But unfortunately, Hadoop doesn't have an in-build authentication capability to authenticate users and propagate their identity. So, the community had following options.

1. Develop an in-build network-based authentication capability into Hadoop.

Manuscript revised on June 19, 2019 and published on August 10, 2019

Dr. Pramod Patil, *HOD & Professor, Dept. of Computer Engineering, DPU, Pune. Email: pdpatiljune@gmail.com*
Dr. Jyoti Rao, *Associate Professor, Dept. of Computer Engineering, DPU, Pune. Email: jyoti.aswale@gmail.com*
Mithun Kankal, *Dept. of Computer Engineering, DPU, Pune. Email: kankal.mithun@gmail.com*

doi: 10.32622/ijrat.77201924

2. Integrate with some other third party system that is purposefully designed to provide the network-based authentication capability.

They decided to go with the second option. So, Hadoop uses Kerberos a third-party tool developed by MIT as part of Athena project for authentication and identity propagation. Kerberos prevent default authentication mechanism in Hadoop and as all machines in the cluster believe every user credentials presented. To overcome this vulnerability the Hadoop uses Kerberos to provide a way for verifying the identity of users. Kerberos authentication and identity verification is implemented via a client/server model as shown in following fig.

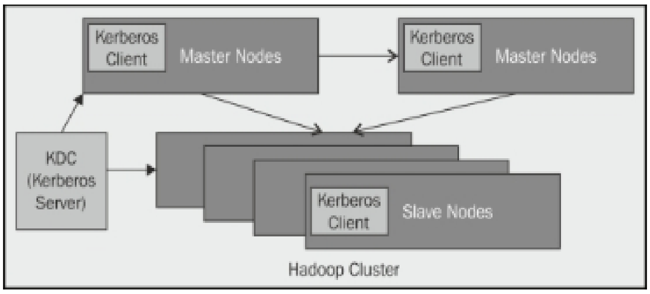


Fig 1: Hadoop Cluster and Kerberos (Narayanan Nov 2013)

The need and use of Big Data/Hadoop have exhibited a wide assortment of security challenges as a result of Kerberos. The following are a few challenges that are shown for Kerberos before a blockchain arrangement is displayed.

- Password-base authentication
- Keys Exposure
- Single Point of Failure
- Time Synchronization
- Denial-of-Service (DoS) Attacks

B. Blockchain

A blockchain, initially block chain, is a persistently developing rundown of records that are connected and verified utilizing cryptography. It is additionally called as Blocks. Each block contains a cryptographic hash of the previous block, transaction information and timestamp of transaction. Blocks hold legitimate transaction data in form of hash and encoded into a Merkle tree.

A blockchain is a distributed, public digital ledger and decentralized and that is used for storing financial transactions record in terms of block across multiple computers which are connected in network so that any involved record which is stored in blocks and placed on computers cannot be altered retrospectively, without the alteration of all blocks stored on multiple computer which are connected in subsequent manner. It allows the participants who are involved in transactions to audit and verify transactions records relativeness in independent manner. A blockchain decentralized database is managed autonomously using a multiple computer which are connected in peer-to-peer network and a distributed timestamping server.

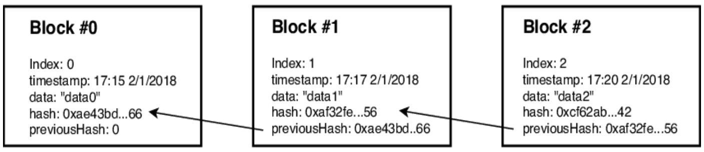


Fig 2: Structure of Generic Blockchain (Kalogeropoulos 2018)

Fig2 presents the structure of generic blockchain. It illustrated blocks include ordered transactions which contains timestamp, data, hash and previousHash. Each block records transactional data and each transaction is linked to the previous one to maintain an ordered structure. Recorded data is information related to anything like user information for authentication of identification of user, banking transaction etc. As a consequence, user can be authenticated against data by verifying information stored in block and transactions can be traced back in time to check authenticity of transaction. In general, a blockchain can possess different characteristics in terms of accessibility. We can use blockchain distributed database in different way based on use case. A classification of these features is presented in the following table as suggested by (Garzik 2015).

Table 1. Designs and characteristics of Blockchain (Garzik 2015)

Blockchain type	Characteristics
Public	No restrictions for reading or submitting transactions for inclusion.
Private	Direct access to data, submitting transactions is limited to a predefined list of entities.
Permission-less	No restrictions on identities of transaction processors.
Permissioned	Solely predefined list of subjects with known identities can process.

C. Consensus Algorithms

For reaching a consensus the Blockchain uses a proof-of-work algorithm. The value must be smaller for cryptographic hash function of each block in order to be considered value. For this a nonce is included in the block. By using the proof-of-work method, in order to change the data in one block, a huge amount of calculation is necessary and all successors of that block must be re-written. In addition, the shorter ones would be discarded at the situation of branches of the chain whereas the longest chain would be accepted by the network. This method or process makes the data in blocks practically unmodifiable or un-hackable, and moreover the harder the processing of overwriting the data where more blocks built upon the block in which the data is contained.

III. BLOCKCHAIN AUTHENTICATION FOR HADOOP

Hadoop is a distributed framework where the primary test is the multifaceted nature of dealing with large implementation, new ways to deal with security are required. Authentication and information access control ought to be overseen by strong authentication, adaptable, versatile and decentralized that denies any vindictive client from gaining admittance to Big Data servers. Henceforth, the new strategy needs to beat the weaknesses of security imperfections in existing execution. This segment quickly talks about the new methodology utilizing blockchain that upgrades verification of Big Data.

The Authentication of Big data using Blockchain is based on the creation of a new HDFS client/gateway interface using existing API of Hadoop ecosystem and new custom python API based on concepts of Blockchain decentralized distributed database. The integration of Hadoop ecosystem and blockchain is a major challenge for this implementation as blockchain is still evolving around the identification and authentication areas.

The HDFS client interface has been build using python along with the creation of blockchain functionality using python libraries. Here, Author uses Private Blockchain type from various available blockchain types for implementation of HDFS Client which will be used for user authentication in Hadoop, where information of a user will be stored as data (transaction) into a block. With the help of basic blockchain features author has created the new authentication mechanism which provides following various features that overcome the shortcoming of Kerberos. The following features described in details in following sub-section.

- Decentralized Authentication
- Unbreakable Record
- Zero Single Point of Failure
- No Session Keys
- Prevent Data Theft

Authentication process has several steps which are described in subsequent sub-section. These steps will extract information of the user from longing detail and verified against the data (transaction) present in blockchain distributed decentralized database. The user will able to access data present on HDFS upon successful verification or authentication process.

Following fig shows how HDFS Client along with blockchain authentication layer would be fit with Hadoop ecosystem.

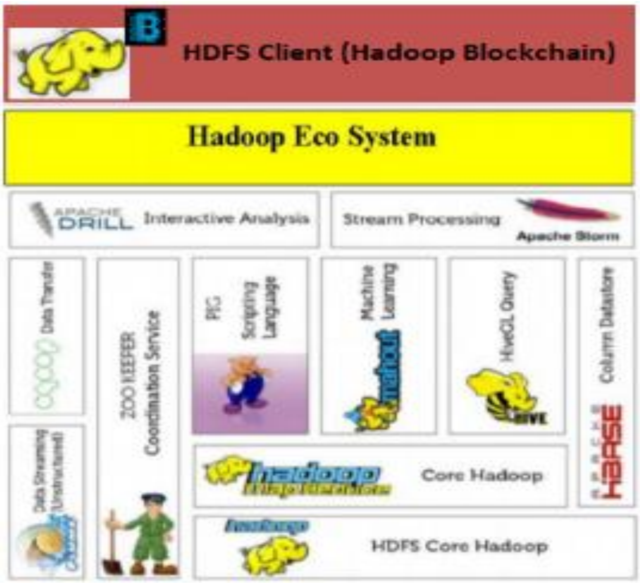


Fig 3: Hadoop and Blockchain (Architecture)

Following are various block chain features that overcomes the shortcoming of Kerberos systems.

A. *Decentralized Authentication*

Decentralized authentication replaces verification instrument which depends on username/secret word created keys and the client side SSL certificate with elliptic bend cryptographic produced keys; this is a similar methodology utilized in a blockchain technology. It removes central databases where user information is stored and managed centrally, which is helpless against programmers who bargain whole accreditations. In this authentication mechanism, the client secret word is just utilized in the client's own machine to get to the private key.

The private key is never moved/uncovered through system or server and can't be traded over a side channel between the server and client. This authentication protocol is based on the digital signature which uses obvious personality confirmation dependent on open keys. A client is confirmed when an exchange or message was appointed by an endorsed private key. It is deduced that the accurate character of the proprietor is unimportant if whoever approaches the private key is the proprietor.

B. *Unbreakable Record*

Blockchain innovation decentralized and disseminated database, which is utilized between gatherings of non-trust parties without need of any center man arrangement or standards to oversee in nonpartisan way, in contrast to Oracle or some other social framework. Blockchain appropriated database that keeps up a chain of record which can develop irreversibly and each record in requested chain rundown named as blocks. Each block contains a Nonce, exchange information and date with time record and a connection to a past information square.

The blocks are inconceivable and unattainable to computationally adjust or change back to past state; it makes attainable to verify block of records from break-in and fake movement. The hash and hash of records makes exchange information can't be adjusted once it is composed to block. The executive or end client of information would not permitted to change or expel any information put away in blocks. Every copy of the record in blockchain in the system must be the equivalent all through the system. At that point, an accord is accomplished by utilizing a proof-of-work convention in the mining procedure. A proof-of-work convention is a bit of information, which is hard to deliver yet simple to check by others client. This makes blockchain distributed database appropriate for account delicate data. For example personality identifiable information, medicinal and financial data.

C. *Zero Single Point of Failure*

Blockchain is a decentralized and distributed database or information stockpiling innovation that keeps up a chain of square or records which persistently develop in an arranged way. It expels the dangers with information which stores halfway and diminishes the defenselessness of single-point disappointment or powerlessness of system programmers.

doi: 10.32622/ijrat.77201924

Each blockchain server or hub which associated in blockchain system contains the duplicate of the blockchain. Nature of information is kept up by enormous replication of database and is cryptographically trusted. Blockchain utilized for client confirmation in a framework makes an un-hackable and sealed advanced character. It possibly diminishes the adequacy of phishing assaults.

The decentralized and conveyed nature of the blockchain system would make it inconceivable for the foundation to flop under an overabundance of solicitations. Henceforth, the verification strategy dependent on blockchain innovation, it is resistant to DDoS assaults and un-hackable.

D. No Session Keys

Utilizing SIN convention is viewed as more secure than session key sharing over the system in a current authentication convention of Kerberos. The SIN can be imparted to everybody straightforwardly, as its relating private key is verified/stored on client side and it never transmitted in a system over the wire, and not imparted to any client or substance. During a confirmation component, the server checks or approve a client by client shared public key against their digital mark and the SIN client shared beforehand. It affirms that SIN past nonce in blockchain square record to avert against replay assaults and in this manner validate the client demand. The advantages of utilizing SIN in recognizable proof instrument is its transportability, where a similar distinguishing proof strategy can be utilized on various gadgets without uncovering clients session key and qualifications over the system.

E. Prevent Data Theft

The expansion in the measure of information burglary and hacking occurrences that have caused shock concerning the getting to of individual sensitive data, specifically money related information, for example, financial balances subtleties, charge cards and wellbeing records or medicinal records. Petland, who are the Professor in MIT has investigated blockchain to construct Enigma, which could possibly permit blockchain conveyed databases to hold sensitive data and procedure it without gambling introduction to malicious programmers. Enigma is depicted as a peer-to-peer network system, empowering distinctive client or associations to together store and run calculations on information while keeping the information totally private. The blockchain innovation makes it harder to break into a framework as opposed to the innovation that does not totally upset robbery. The total usage and framework of blockchain upgrades protection, opportunity and security of transport of information.

IV. IMPLEMENTATION DETAILS

The blockchain generated data for authentication is secure because of blockchain network architecture and due to which it cannot be forged or modified. In a blockchain, data is completely transparent as if a record is not verified then it is automatically rejected. The author has used private blockchain type mechanism for implementation of an authentication process in the form of HDFS Client. The Private Blockchain is also called as permissioned blockchain and in this type of blockchain network has a restriction on who is allowed to participate in the financial transaction record as participant or any other transaction which are used record user information. In private blockchain, we usually know who are the individual users are, and from which origination they are, and what role they are associated with. Here we also assume that user behaves fairly and misbehaving in any way, they have gone suffer the consequences for that.

Following are some of the benefits of private blockchain are.

- Enterprise or Single Organization and Permissioned.
- Identities are Known
- Lighter Blockchain and Faster Transaction Speed.
- Participants are Pre-approved.
- Better Scalability.
- More Efficient Consensus Process.

Using blockchain for Authentication process in form of HDFS Client adds additional layer to big data analytics process. This additional layer of authentication for Hadoop compiles of two main modules which are as follows.

- Blockchain Admin Module.
- Blockchain Authentication Module using HDFS Client.

A. Data Flow Diagram

The High level functionality of these is represented in following Data flow diagram of system.

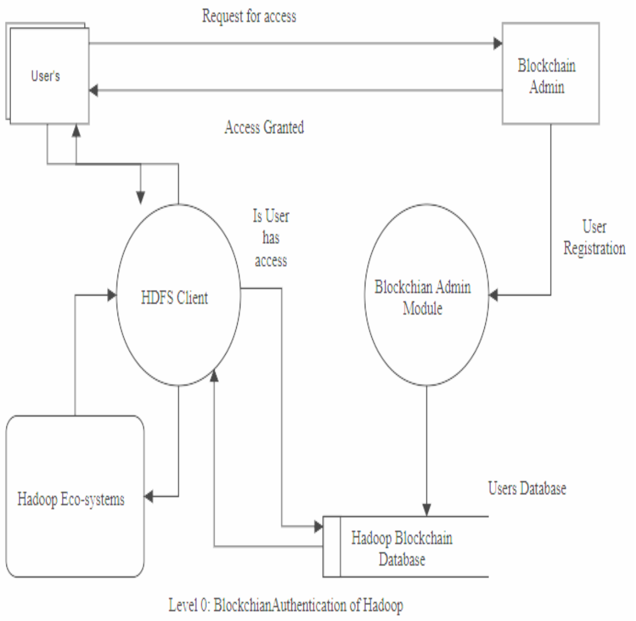


Fig 4: Level 0 Data Flow Diagram

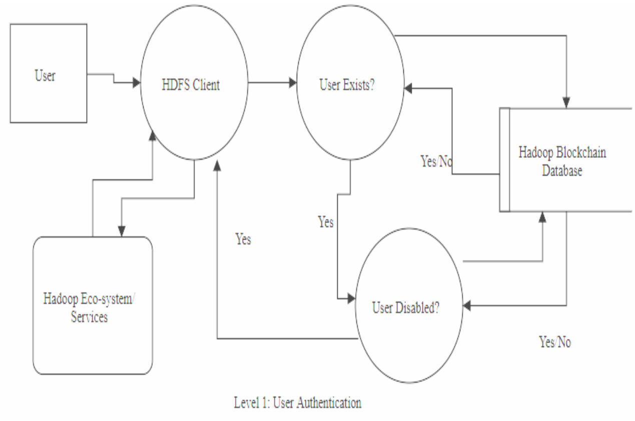
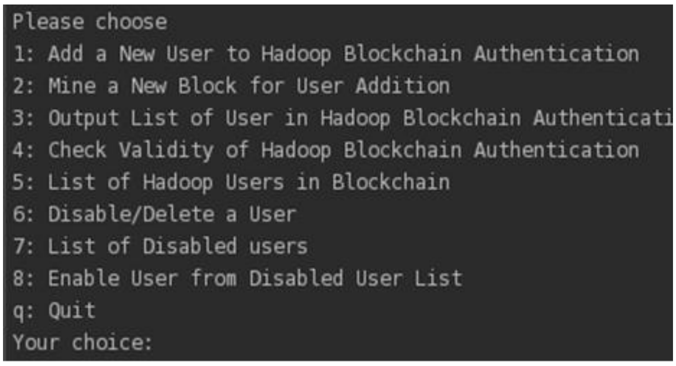

User
HDFS Client
User Exists?
Hadoop Blockchain Database
Yes/No
Yes
Yes
Hadoop Eco-system Services
User Disabled?
Yes/No
Level 1: User Authentication

Fig 5: Level 1 User Authentication DFD

V. Result And Discussion

A. Blockchain Admin Module

The Author has implemented the Blockchain Admin Module as part of Private Blockchain Permissioned network, where user can participate in blockchain and access the HDFS file system only after completion of pre-approved user access provisioning process. Following are various operational features implemented and provided for blockchain admin module that are shown in following Fig.



```
Please choose
1: Add a New User to Hadoop Blockchain Authentication
2: Mine a New Block for User Addition
3: Output List of User in Hadoop Blockchain Authenticati
4: Check Validity of Hadoop Blockchain Authentication
5: List of Hadoop Users in Blockchain
6: Disable/Delete a User
7: List of Disabled users
8: Enable User from Disabled User List
q: Quit
Your choice:
```

Fig 6: Blockchain Admin Module Options.

- ***User Registration in Blockchain Database***

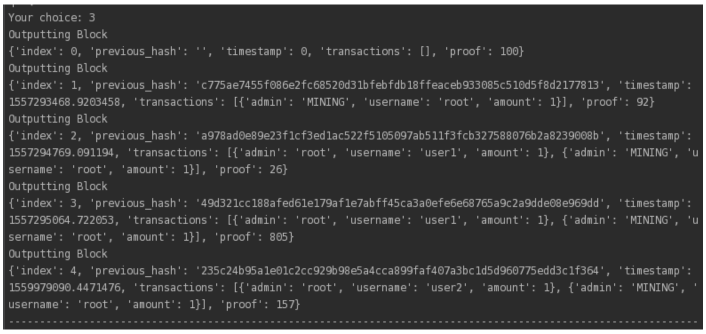
This operational feature used for adding new user to blockchain database for Hadoop authentication. However the user will be added temporarily till the newly added user is not validated and verified by Mining Operation. For adding user permanently blockchain admin has to execute operation 1 followed by Operation 2. If process of adding new user failed then blockchain admin has to do mining before adding any new user. Here each user information is stored in block, each block contain 64 bit hash code of previous block along with transaction data. The transaction data will contain the information about user.

- ***Blockchain Mining for User Access Provisioning***

This operational feature to verify existing blockchain and validates it contents before adding any new user information to existing blockchain. The new user information is added to blockchain upon successful validation of blockchain. This operation takes consensus from other participating node in blockchain before adding new user information into new block at the end of blockchain.

- ***Displaying Blocks of Blockchain***

This feature will display the list of users exists in Hadoop blockchain database in form of chain of block with hash code of each block along with user information. Here, each row consists of one block from blockchain which is represented by index value 0, 1, 2 etc. This index value will be incremented after new block added to blockchain db. The Index 0 block represent the genesis block and remaining following block contains user information along with 64bit hash code of previous block.



```
Your choice: 3
Outputting Block
{'index': 0, 'previous_hash': '', 'timestamp': 0, 'transactions': [], 'proof': 100}
Outputting Block
{'index': 1, 'previous_hash': 'c775ae7455f086e2fc68520d31bfebfdb18ffeaceb933085c510d5f8d2177813', 'timestamp':
1557293468.9203458, 'transactions': [{'admin': 'MINING', 'username': 'root', 'amount': 1}], 'proof': 92}
Outputting Block
{'index': 2, 'previous_hash': 'a978ad0e89e23f1cf3ed1ac522f5105097ab511f3fcb327588076b2a8239008b', 'timestamp':
1557294769.091194, 'transactions': [{'admin': 'root', 'username': 'user1', 'amount': 1}, {'admin': 'MINING', 'u
sername': 'root', 'amount': 1}], 'proof': 26}
Outputting Block
{'index': 3, 'previous_hash': '49d321cc188afed61e179af1e7abff45ca3a0efe6e68765a9c2a9dde08e969dd', 'timestamp':
1557295064.722053, 'transactions': [{'admin': 'root', 'username': 'user1', 'amount': 1}, {'admin': 'MINING', 'u
sername': 'root', 'amount': 1}], 'proof': 805}
Outputting Block
{'index': 4, 'previous_hash': '235c24b95a1e01c2cc929b98e5a4cca899faf407a3bc1d5d960775edd3c1f364', 'timestamp':
1559979090.4471476, 'transactions': [{'admin': 'root', 'username': 'user2', 'amount': 1}, {'admin': 'MINING', '
username': 'root', 'amount': 1}], 'proof': 157}
-----------------------------------------------------------------------------------------------------------------
```

Fig 7: Displaying Blocks of Blockchain

- ***Validation of Blockchain***

The validation feature gives blockchain admin to verify the entries in blockchain database are in intact and no one has done any tampering or modification to it. The validation operation is lightweight operation as compared to Mining process as it will take less resources. This process will only passed when the blockchain hash code present in current block is matched with newly calculated hash of previous block in blockchain. If this operation is failed then the node on which this operation failed is considered as faulty node. If any node declared as faulty then all operational attempt to access data on HDFS will be failed. Also the validation of blockchain process will fail if someone has modified or try to modify entries in blockchain.

- ***Listing Users from Blockchain Database***

This feature will shows the list of users exists in blockchain database which are authenticated to access Hadoop using HDFS Client.

- ***Disabling Hadoop User***

This operational feature allows the blockchain admin to disable the Hadoop access of user for any reason. Such that user would not able to access the HDFS file system using authentication module.

- ***Displaying Disabled Hadoop User List***

This operational feature allows blockchain admin to check/displays the list of user whose access have been disabled and user cannot access Hadoop by any way using HDFS Client.

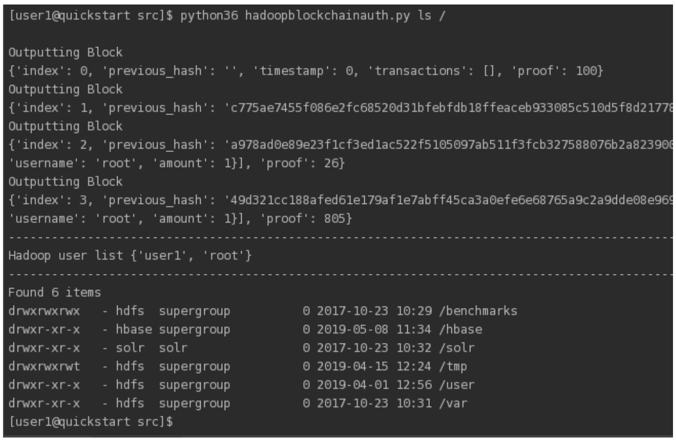
- ***Enabling User***

doi: 10.32622/ijrat.77201924

This operational features allows blockchain admin to enable Hadoop access of user from disabled user list.

B. Blockchain Authentication Module using HDFS Client

The Author has implemented this Blockchain Authentication Module which consists of two parts. First part is private blockchain, where user can participate in blockchain and access the HDFS file system only after pre-approval and user access provisioning process. Second part is HDFS Client which contains interface to HDFS and interact using command line interface. Following fig shows how to use blockchain based HDFS Client to access the HDFS directory and successful user authentication using blockchain.

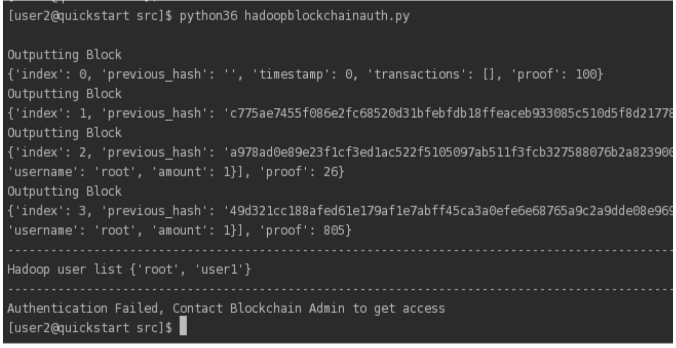


```
[user1@quickstart src]$ python36 hadoopblockchainauth.py ls /

Outputting Block
{'index': 0, 'previous_hash': '', 'timestamp': 0, 'transactions': [], 'proof': 100}
Outputting Block
{'index': 1, 'previous_hash': 'c775ae7455f086e2fc68520d31bfebfdb18ffeaceb933085c510d5f8d21778
Outputting Block
{'index': 2, 'previous_hash': 'a978ad0e89e23f1cf3ed1ac522f5105097ab511f3fcb327588076b2a823900
'username': 'root', 'amount': 1}], 'proof': 26}
Outputting Block
{'index': 3, 'previous_hash': '49d321cc188afed61e179af1e7abff45ca3a0efe6e68765a9c2a9dde08e969
'username': 'root', 'amount': 1}], 'proof': 805}
-------------------------------------------------------------------------------------------
Hadoop user list {'user1', 'root'}
-------------------------------------------------------------------------------------------
Found 6 items
drwxrwxrwx   - hdfs  supergroup          0 2017-10-23 10:29 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2019-05-08 11:34 /hbase
drwxr-xr-x   - solr  solr                0 2017-10-23 10:32 /solr
drwxrwxrwt   - hdfs  supergroup          0 2019-04-15 12:24 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2019-04-01 12:56 /user
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 10:31 /var
[user1@quickstart src]$
```

Fig 8: Successful User Authentication using Blockchain

Here, for example, the user2 try to access the Hadoop but it will not able to access it because user2 does not exists in blockchain authentication database and get authentication failed message.



```
[user2@quickstart src]$ python36 hadoopblockchainauth.py

Outputting Block
{'index': 0, 'previous_hash': '', 'timestamp': 0, 'transactions': [], 'proof': 100}
Outputting Block
{'index': 1, 'previous_hash': 'c775ae7455f086e2fc68520d31bfebfdb18ffeaceb933085c510d5f8d21778
Outputting Block
{'index': 2, 'previous_hash': 'a978ad0e89e23f1cf3ed1ac522f5105097ab511f3fcb327588076b2a823900
'username': 'root', 'amount': 1}], 'proof': 26}
Outputting Block
{'index': 3, 'previous_hash': '49d321cc188afed61e179af1e7abff45ca3a0efe6e68765a9c2a9dde08e969
'username': 'root', 'amount': 1}], 'proof': 805}
-------------------------------------------------------------------------------------------
Hadoop user list {'root', 'user1'}
-------------------------------------------------------------------------------------------
Authentication Failed, Contact Blockchain Admin to get access
[user2@quickstart src]$ █
```

Fig 9: Unsuccessful User Authentication using Blockchain

C. Comparison:

Table 2. Comparison between Keberos and Blockchain Auth.

	Kerberos	**Blockchain Authentication**
Authentication Type	Centralized	Decentralized
Authentication Mechanism	Password based	Password less
Session Key	Time based session Key authentication	No Session Key
Failure	Single Point Failure	Decentralized
Exposure to attacks	Brute Force, DDoS etc.	Unbreakable/ un-hackable

VI. CONCLUSION AND FUTURE SCOPE

Here author represents the common security problems associated with Kerberos. The Kerberos uses in large network such as the internet and increasingly used in variety of systems such as Big Data environment where security vulnerability is common and it is highly vulnerable due to shortcoming of Kerberos. Here the Kerberos limitations have been addressed.

New solutions would be needed for big data environment in an era where greater security requirements needed as integration of big data system happening with different other system. Blockchain technology has provided various scalable security solutions to for various fields in multiple sector, and it was first introduced by Bitcoin, also it is been used in researching for solving other various common security issues. We also tried to use blockchain technology to build authentication mechanism for Hadoop.

Existing authentication mechanism using Kerberos, positions Big Data systems to depend on many security risks and vulnerabilities. The mechanism for Hadoop authentication using blockchain is based on distributed and decentralized infrastructure and that is scalable, reliable and has no single point failure.

Therefore, the utilizing the advantages of blockchain technology could be leveraged to harden security systems, including distributed authentication and no single point failure of Big Data system and there is no failure due to centralized servers as the mechanism is based on distributed technique. Hence author tried to build a new identity system and authentication framework for big data in form of HDFS Client which is based on blockchain technology. This authentication mechanism is built with cloudera quickstart vm along with python libraries which are used to create blockchain. Currently this mechanism is works for one node and in future work we can extends and build it for Hadoop cluster which has multiple hosts in it.

REFERENCES

[1] Nazri Abdullah, Anne Håkansson, Esmiralda Moradian "*Blockchain based Approach to Enhance Big Data Authentication in Distributed Environment*" In: International Conference on Ubiquitous and Future Networks. ICUFN, pp. 887–892.

[2] Mithun Kankal, Pramod Patil "*An Adaptive Authentication Based on Blockchain for Bigdata Hadoop Framework*" Volume 5 - Issue 1 (89-94) January - February 2019, International Journal of Engineering and Techniques (IJET),ISSN:2395-1303, www.ijetjournal.org

[3] "*Cloud Security Alliance and CSA Releases the Expanded Top Ten Big Data Security & Privacy Challenges:*" [Online] at: https://cloudsecurityalliance.org/media/news/csa-releases-theexpanded- Top-ten-big-data-security-privacy-challenges/. [Accessed: 19-Jan-2016].

[4] *"Welcome to ApacheTM Hadoop®!"* [Online]. Available: https://hadoop.apache.org/. [Accessed: 12-Jan-2016].

[5] S. M. Bellovin and M. Merritt, "Limitations of the Kerberos Authentication System," SIGCOMM Compute Common Rev, vol. 20, no. 5, pp. 119–132, Oct. 1990.

[6] D. Davis and D. E. Geer, *"Kerberos Security with Clocks Adrift."* in USENIX Security, 1995.

doi: 10.32622/ijrat.77201924

[7] D. E. Denning and G. M. Sacco, *"Distribution Protocols and used of Timestamps,"* Commun ACM, vol. 24, no. 8, pp. 533–536, Aug. 1981.

[8] "Intel-hadoop/project-rhino," GitHub. [Online]. Available: https://github.com/intel-hadoop/project-rhino. [Accessed: 23-Mar-2016].

[9] [Online] Available: *"Lightweight Directory client server Access Protocol," Wikipedia, the free encyclopedia. 20-Mar-2016.*

[10] [Online] Available: https://101blockchains.com/consensus-algorithms-blockchain/

AUTHORS PROFILE

Mithun Kankal
Hadoop Administrator and Developer with 9+ years of Industry Experience on big data and big data analytics using Hadoop and hadoop ecosystem. Working as Researcher and Implementer of various new technologies like Cloudera, AWS, Ansible, Teraform etc.

Dr. Pramod Patil
An alumnus of COEP Pune, Pramod holds Masters in Computer Engineering and Ph.D from COEP. He has total 14 years of experience in Academics, Research and Industry. He held various positions such as HOD, Associate Professor, Assistant Professor, and Lecturer during his tenure. He is recognized as a Post Graduate Teacher, Computer Engineering at University of Pune.

Dr. Jyoti Rao
Total 12.5 years of Teaching Experience in Computer Engineering in DYPIET Pimpri Pune 18. PhD in Vignan University, Guntur, Andhra Pradesh. Approved PG Teacher. Development proficiency in Microsoft Technologies C++, C Sharp. Have developed a live project for BMC Software Pune during 2007-2008. Worked on Unix, Solaris, Linux in IUCAA during 2000 – 2001.

doi: 10.32622/ijrat.77201924