Integrated HDFS for secure traceability With Blockchain

Ms. Reena Sharad More
Department of Information Technology
MVPS's RSM Polytechnic
Nashik, India
reena.more@rsmpoly.org

Ms. Dipika Laxman Tidke
Department Computer Engineering
MVPS's KBT COE
Nashik, India
dipikatidke@kbtcoe.org

Ms. Madhuri Vishnu Malode
Department Computer Engineering
MVPS's KBT COE
Nashik, India
malode.madhuri@kbtcoe.org

***Abstract*— Hadoop Distributed file system (HDFS) is one amongst the widely used distributed file systems in huge information analysis for frameworks like Hadoop. HDFS permits one to manage giant volumes of knowledge and mistreatment of inexpensive goods hardware. However, vulnerabilities in HDFS may be exploited for wicked activities. This reinforces the importance of guaranteeing sturdy security to facilitate file sharing in Hadoop further as having a trusty mechanism to check the credibility of shared files. This is often the main focus of this paper, wherever we have a tendency to aim to enhance the safety of HDFS employing a blockchain-enabled approach (hereafter brought up as BlockHDFS). Specifically, the projected BlockHDFS uses the enterprise-level Hyperledger cloth platform to exploit file information for building trusted information security and traceability in HDFS.**

Keywords- HDFS, Integrated Hadoop framework with Blockchain, security and traceability in HDFS.

I. INTRODUCTION

Hadoop distributed File system (HDFS) [1,2], one among the foremost widely used file systems in organizations that touch upon huge knowledge applications, is mainly used for instruction execution of information. it's renowned for its high throughput knowledge accessibility with low latency. There are many alternatives to HDFS, like Ceph [3], GPFS [4], and Hydra [5]. In contrast to alternative file systems, HDFS was developed by the Apache code Foundation with the particular aim to store massive datasets in an exceedingly distributed manner to address hardware failures and to perform Map cut back [6] operations for data analytics. MapReduce could be a pre-built framework in HDFS. The users can use one or a lot of files in MapReduce to map and cut back (sort) the info inside files to get the specified output. Java code snippets are, however, required to be written to perform MapReduce operations. MapReduce considers a get in the input directory, and therefore the desired output is created to write in the associate output directory. What makes HDFS thus common nowadays is, perhaps, its design. HDFS is comparable to UNIX system classification system design, however the

info is distributed among many exhausting disks and may be run on trade goods hardware, creating it cheaper to use and maintain. Apache Hadoop is an ASCII text file framework that runs on high of HDFS. Being a distributed classification system, Apache Hadoop is effective. Right from the start, the motivation of Hadoop has been to create a high throughput system with fail-safe options. There are 2 ways in which to transfer a file into HDFS, namely: through the program line and victimization genus Apis. Both ways are accessed by victimization SSH if there's a requirement for an overseas affiliation, and the program line and API have multiple commands for various file operations. HDFS supports making, appending, opening, deleting, renaming, status, and listing functions. Once a file is uploaded into HDFS, various system frameworks (e.gSpark [7], Hive [8], HBase [9]) will be accustomed to perform the specified knowledge analytics. These system applications typically use the Hadoop Main API to perform knowledge operations from and to the HDFS. Inspired by the distinctive security-by-design and tamper-resistant characteristics of blockchain technology in up to date application domains [10–14], we tend to propose a brand new approach, referred to as BlockHDFS, that incorporates blockchain functions into the HDFS by capitalizing on file metadata keep in tamper-proof blocks. In our work, we tend to use Hyperledger Fabric [15] to induce the data from HDFS and store them within Hyperledger as associate plus. Hyperledger cloth could be a permissioned blockchain with a distributed ledger that supports good contracts [16]. solely attested users are able to see the info within Hyperledger, in contrast to alternative public blockchains like Ethereum, Bitcoin, or Monero. victimization Hyperledger cloth additionally needs less computation power to run, as everything resides on a non-public business network fitted to organizations and enterprise needs. Consequently, this eliminates the requirement of blockchain mining. Storing data of the files into the blockchain is enforced on virtually each file system as each classification system must maintain data. HDFS uses CRC32 or CRC32C, a 32-bit Cyclic Redundancy Check based on the Castagnoli polynomial algorithmic rule to get the hash of the files. However, once identical hash is accessed through Hadoop's API, the MD5 version of hash is created, that is MD5-of-MD5-of-CRC32C. This hash is accustomed to check if there's a modification within the file when it's kept in HDFS. To the most effective of our information, our work is one among the primary makes an attempt in the analysis community to integrate blockchain into HDFS for higher data security and trustworthy traceability. Overall, our results indicate positive impacts on the protection and traceability of files in HDFS, which makes the projected approach promising and noteworthy. The rest of this paper is structured as follows. Section two introduces the reader to the specified background and provides an outline of the related approaches. In Section three, the threat model is given. Section 4 presents the projected BlockHDFS.

II. BACKGROUND AND CONNECTED WORK

2.1. Hadoop Distributed file system (HDFS)

HDFS [2], a distributed file system for parallel massive processing, uses the master-slave design to schedule the tasks and manage the data transfer among totally different computing nodes. In HDFS, the master node is named NameNode and also the slave node is named DataNode. Multiple NameNodes will be accustomed to maintaining high handiness victimization Associate in Nursing active-passive relationship. NameNodes are the unit to blame for playing block operations, and that they store information associated with the filing system, such as wherever every chunk of a file is found. HDFS has been widely employed by distributed processing frameworks like Map scale back [6] to perform economical massive information analytics jobs. A MapReduce job consists of Map tasks that perform identical map operation on totally different information chunks in parallel and scale back tasks that use shuffling and reducing functions to generate desired outputs. totally different MapReduce jobs will share identical clusters of machines and area units coordinated by employment huntsmen. HDFS will be enforced on goods hardware with good computing power. HDFS maintains replicas of a go into the shape of blocks. The blocks are a unit gift in several DataNodes so as to recover files in case of a crash of a DataNode, that makes HDFS extremely fault-tolerant. Fig. 1 shows the scan and write operations performed on HDFS area units collaboratively managed by NameNodes and DataNodes. in addition, NameNodes maintain logs of all the operations and store them within a picture file at checkpoints. A stop may be a time wherever all the logs are combined into a file referred to as fsImage. Once a NameNode is turned back on once being switched off, all the data regarding the blocks is loaded from the fsImage file.

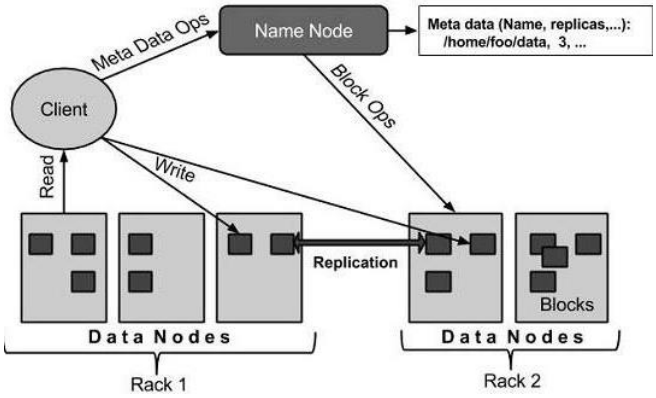

Fig 1. Architecture of HDFS

2.2 *Hyperledger cloth*

Hyperledger cloth [15] is Associate in Nursing ASCII text file and brazenly ruled collaborative effort to advance cross-industry blockchain technologies for business and also the UNIX Foundation hosts it. Hyperledger cloth may be a blockchain framework implementation and one in all the Hyperledger projects supposed as a foundation for developing applications/solutions with a standard design fitted to enterprise-level solutions. The main parts of Hyperledger cloth area unit shortly delineate below.

2.2.1. Distributed ledger

The Fabric ledger has 2 parts: State information and dealings Logs. The ledger is that the sequenced, tamper-resistant records of all state

transitions. State transitions area unit the results of chain code invocations (“transactions”) submitted by taking part parties. every dealings leads to a collection of asset key-value pairs that area unit committed to the ledger as creates, updates, or deletes. Every deal features a distinctive ID, its time-stamp, and contains signatures of each endorsing peer, and is submitted to the ordering service. The ledger is comprised of a blockchain to store the immutable, sequenced records in blocks, similarly as a state information to take care of the current cloth state. There’s one ledger per channel. Every peer maintains a copy of the ledger for every channel within which they're a member.

2.2.2. Nodes

The idea of ‘node’ is common altogether with blockchain technologies. In a public blockchain like Ethereum [17], anyone will participate as a node by downloading the node consumer software package. A node becomes the communication end point within the blockchain network. Peer-to-peer protocols are units used to keep the distributed ledger in synchronization across totally different nodes within the network. In an exceedingly permissioned blockchain, like Hyperledger cloth, nodes would like a sound certificate to be ready to communicate to the network, where the participant's identity isn't identical because of the node's identity.

2.2.3. Channel

Members will participate in multiple Hyperledger blockchain networks. dealings in every network are isolated and this can be created by the manner of what's cited because of the channels. Channel may be an information partitioning mechanism to manage dealings visibility solely to stakeholders. Non-registered members on the network don't seem to be allowed to access the channel and cannot see transactions on the channel.

2.3. Related work

There is a unit of many works [18–20] that debate the combination of distributed massive information platforms and blockchain in literature. The main intention of HDFS is to accelerate massive processing. Therefore, very few safety features were embedded in HDFS at its early stage. Later on, Kerberos Authentication Protocol [21] was created to boost HDFS authentication.For instance, HDFS began to support ACL (Access management List) for file authorizations. It conjointly supports auditing for system security [22], which, however, lacks a format to create such auditing straightforward to read or be unified. Some ecosystems running on HDFS, like Apache NiFi [23], network auditing additional profitably, however there's no direct thanks to find of the audit logs in an exceedingly normal format. Coming to the main security feature in HDFS, that is coding, most of the info sent between the HDFS and its purchasers is completed through RPC (Remote Procedure Call), that is encrypted victimization SASL (Simple Authentication and Security Layer) RPC. There are 2 sorts of information encryptions in HDFS which are unit data-at-rest and data-in-transit. Data-in-transit refers to the data being transmitted

from the purchasers to HDFS, and it will be encrypted using RPC, as mentioned above. However, data-at-rest that refers to the data held in HDFS, remains at high risk. There are only a few approaches to secure information at rest. Project rhinoceros [24], for instance, is designed to assist write the data-at-rest, however this system needs tweaking the parts of HDFS like configuration files, adding new properties, and ever-changing ports because it is put in system-wide. Orthogonal to the approach projected by rhinoceros [24], our answer targets providing Associate in Nursing immutable set of logs transparently. This approach does not need any amendment to the user application code. While such immutable logs will function as a sure supply of data to assist users investigate what has gone wrong once the info in HDFS is compromised.

III. THREAT MODEL

Many hackers have targeted HDFS in the main attributable to the values of the data it holds. The data in HDFS is usually structured data that contains a lot of information, like sales, employees, and analytical market data. Fig.2 shows projected threat model in HDFS. The line in this figure represents the Trust Boundary. On the left of the line is that the users, UN agency, hook up with the HDFS that are on the proper facet of the dotted V. Mothukuri et al. Blockchain: analysis and Applications a pair of (2021) 1000322 lines, via the API element. Samples of such API parts are often WebHDFS [25], or HttpFS [26], that are chargeable for all the information retrieval and modifications. Because of approaches like SASL RPC, data being uploaded to HDFS is tough to be altered before it reaches HDFS. Concretely speaking, Hadoop uses open ports to project the API to the Internet or a network. Each open port has its own purpose. As an example, open port twenty three is employed for telnet services. Hackers will exploit such open ports to perform vulnerability scans that eventually permit them to access the information within the DataNodes lawlessly. However, for knowledge on HDFS, a hacker will exploit the API and modify the information within the DataNodes. In most cases, the assaulter performs the footprinting, mistreatment computer code like NMap [27] and Zen Map [28], to urge the IP address of the target network and then carries out port scanning to search out data like the operating systems, the running services, or system design of the machines on this network. Since an associate degree open port with a method running in the background can invariably listen for input from alternative processes within the network, hackers can notice these processes and exploit them. For example, WebHDFS uses a default port of 50070 for NameNode, and organizations use this port by default. A malicious user UN agency gains access to such a port can simply modify the information within the DataNode, and everyone it takes is that the name of the user UN agency is mistreating the HDFS.

To tackle such attacks, our answer takes the advantage of blockchain to create an associate

degree changeless set of logs for HDFS, which may function as a sure source of knowledge for a good investigation once one thing has gone wrong in HDFS. Although our answer cannot utterly stop a hacker from assaulting the HDFS knowledge via associate degree open port, it provides the simplest way to log the information modifications during a permanent and tamper-proof method, which helps to guide the investigators to spot the hackers.

IV. BLOCKHDFS

In this section, we introduce our proposed solution named BlockHDFS, which enhances the security of HDFS in a user transparent way by storing metadata of files from HDFS in a blockchain.

4.1. Functional Design of BlockHDFS

Fig.2 presents the planned design of BlockHDFS that consists of 3 components: associate degree HDFS cluster together with the NameNodes and DataNodes, a permissioned blockchain network like Hyperledger Fabric, and a NodeJS consumer that acts as a bridge connecting the HDFS cluster and also the blockchain network.

BlockHDFS is meant in a very user clear manner, so it shares several features and operations with ancient HDFS. As an example, a user uploads a file to BlockHDFS by mistreating either the API or statement, and the file transfer is secured by mistreatment DEK coding [29] once information is at rest. The file residing within the DataNode is then replicated into a variety of other DataNodes supporting the predetermined replication issue. In our style, Hyperledger material [30] is employed to store file information, which however is not restricted to the hash worth, interval, and modification time of a file into the blockchain, whereas HDFS still manages the files themselves. In addition, a NodeJS consumer sporadically extracts the file information from HDFS mistreatment the WebHDFS REST API and so stores them within the ledger of Hyperledger material. The user will forever question the Hyperledger material to understand what went on to a given file, as all the connected info is held on the blockchain ledger, which provides associate degree changeless and trustworthy traceability.

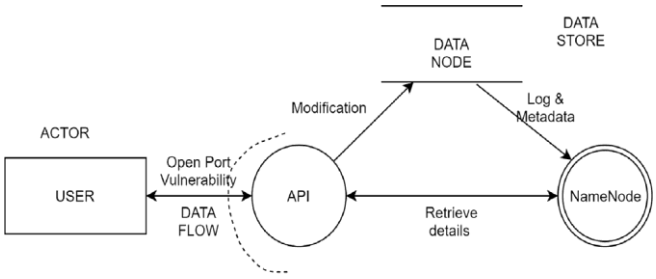

Fig. 2 Data flow diagram of threat model.

It is worth noting that the consumer in Hyperledger material just wants to store the file information, additionally called information together with interval, modification time, and hash value—note, this design permits adding multiple information on the far side those listed here.The files themselves are still managed by HDFS itself. The file information is provided by the WebHDFS REST API. The consumer in between is liable for obtaining these triplet values

from the remainder API and storing them in Hyperledger Fabric. The user will forever build a question to the Hyperledger material to read changes that happened to a given file as all the knowledge is hold on in the style of secured transactions on the blockchain, rising security and traceability.

In BlockHDFS, the blockchain is liable for storing the information of the files. Overheads incurred by storing the HDFS file information into the blockchain square measure from 2 aspects. First, the WebHDFS REST API desires to scan the information, like the hash worth, of a file from HDFS. Retrieving file information from HDFS will introduce some latency. However, as is shown within the experiments, such latency isn't vital. Second, it takes extra operations to store the information into the blockchain. However, since information size is typically tiny, such overhead will neither introduce high latency for HDFS operations nor need a large amount of storage on the blockchain. Hyperledger consumer part is employed to speak between Hyperledger material networks and enable the interaction with the ledger. Once a file is added to the file system, the consumer can get the computer file name mistreatment LIST A DIRECTORY operation and retrieve information, like the hash worth, interval, and modification time of a file, from HDFS mistreatment WebHDFS API. Then the JSON information is side to Hyperledger mistreatment of the NodeJS consumer. The NodeJS client can inform users just in case of failing to write down the information into the blockchain.

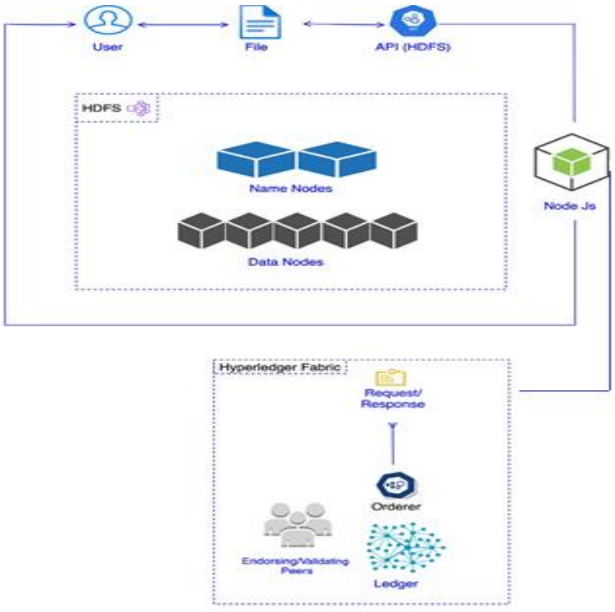

Fig.3 Architecture of BlockHDFS

Following, we tend to describe most parts of BlockHDFS and its functions in reaching our security goals.

4.1.1. User

In BlockHDFS, a user is taken into account AN administrator or one among the employees of a company World Health Organization has already been given privileges and access to the HDFS. A user will add, modify, and delete

the information within HDFS supported ACL rules. as an example, victimization the WebHDFS API, a user can add a file to HDFS as follows:

curl -i -X PUT "http://:
/webhdfs/v1/? op¼CREATE"

To add, modify, or delete a file, a user won't bear in mind the existence of the Hyperledger material.

All the file operations are performed using Hadoop's API solely. The user won't directly move with the Hyperledger unless she/he desires to see the data on the blockchain.

4.1.2. WebHDFS REST API

This REST API is wont to perform file operations and notice data concerning the files in HDFS. If the protection is turned off on an area server by disabling the Kerberos or by turning Safe Mode off, the REST API operations are performed by anyone on the network while not authentication. If the protection is turned on, then the authentication is taken care of by Kerberos [21]. Authentication victimization SSH [31] is required to access Hadoop API if the NodeJS shopper is hosted on a foreign server instead of the native Hadoop server.

4.1.3. File confirmation

HDFS uses the CRC32 hashing rule [32], whereas the confirmation produced by WebHDFS REST API is that the MD5 of CRC32. Once a user uploads a file to HDFS, she/he ought to check the credibility of the file by validating if the confirmation in HDFS matches the confirmation on her/his native machine. Albeit the file transfer is secured by cryptography in RPC [33], it's typically useful to verify if the file is uploaded properly without any errors. Therefore, such validation is enclosed in BlockHDFS, which gets the confirmation of a file victimization REST API by victimization the below

command:

curl -i -X PUT "http://:

/webhdfs/v1/? op¼GETFILECHECKSUM"

4.1.4. NodeJS shopper

The NodeJS shopper is AN application that wants to get file related information, like confirmation, interval, and modification time of a file, from HDFS. This application additionally sends the information received from HDFS to the Hyperledger material. In observe, this application is written in any language that supports REST API operations, while we used NodeJS. If this NodeJS shopper runs on a foreign server, authentication is required to access the WebHDFS REST API server.

4.1.5. Hyperledger material blockchain

Hyperledger material could be a permissioned blockchain appropriate for enterprise applications, which needs user authentication. In BlockHDFS, only users of HDFS are going to be given access to Hyperledger material, which reduces the danger of exposing the blockchain ledger to unauthorized users. HDFS file information sent by NodeJS shoppers, like file name, hash value, access time, and modification time of a file, is held on within the blockchain ledger within the type of assets. The ledger is used as AN changeless and trusted supply of logs by the user if she/he suspects one thing has gone wrong with the file, that provides a straightforward thanks to trace what has happened to a file over time.

4.2. Implementation of BlockHDFS

The consumer application in between the HDFS and therefore the Hyperledger material is enforced

in NodeJS. we have a tendency to use the Hadoop WebHDFS REST API consumer library for NodeJS to support asynchronous functions, rising the code's performance and responsiveness by sanctionative parallel process.we have a tendency to ar employing a consumer library referred to as webhdfs-npm, that supports numerous numerous system functions. Here within the code, we are able to use asynchronous functions to execute the WebHDFS API functions quicker while not looking forward to different functions. Once a file is uploaded to the HDFS, the NodeJS consumer checks if a replacement a replacement been uploaded to the HDFS victimisation WebHDFS REST API by capital punishment the LISTDIRECTORY operation. If there's a replacement file, then the NodeJS consumer retrieves the hash price of this file victimization GETFILECHECKSUM yet because the interval interval time of this file victimization GETFILESTATUS. All such information is parsed from a JSON format. The NodeJS consumer then writes these values into the Hyperledger material as Associate in Nursing plus.

Fig. 4 WebHDFS JSON data

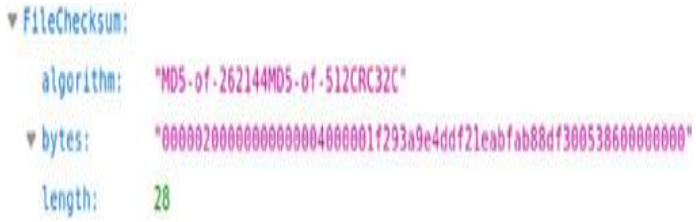

Fig. 5.WebHDFS checksum data.

Fig.6 Filedata in blockchain.

Hyperledger material v1.1 and Hadoop v3.1.2 are employed in the paradigm. A script was developed to automatise all the tasks required to urge the BlockHDFS up and running. As an Associate in Nursing example of how BlockHDFS works, 3 3 of various sizes are uploaded onto HDFS. The MD5 hash values of those generated by WebHDFS are written into the Hyperledger material employing a REST API. The NodeJS consumer is answerable for causing the GET decision to WebHDFS and place response to the Hyper- ledger material. Fig.4 shows what the JSON information from WebHDFS is like, wherever the directory structure in conjunction with and in conjunction with and different information may be seen. Such information may be received by causing the LISTDIRECTORY API decision to the WebHDFS. Fig.6 shows the confirmation JSON information from one among the folders within HDFS. This piece of information comes from WebHDFS. As mentioned earlier, the data of a file is distributed to the Hyperledger material as Associate in Nursing plus with the assistance of the NodeJS consumer. An Associate in Nursing example of such information held within the blockchain is shown in Fig.6. If there's a modification to the current modification, a replacement plus are going to be further in conjunction with the date and time by the NodeJS consumer. Such assets on the blockchain may be used as Associate in Nursing immutable and sure log by the user to trace the history info of a file. In our style, the NodeJS consumer checks the file changes periodically employing a feature referred to as CRON JOBS [34] in the UNIX system. Fig.6 shows the info within the blockchain ledger in JSON format. The plus Id field contains each hash price and modification time of a file, whereas the worth(Note that the particular length of the hash value was cut to form into the figure).

LIMITATIONS

Although the BlockHDFS includes a generic and extremely climbable design, its current implementation has some limitations. First, it's potential to create folders and sub-folders within HDFS to store files. The depth of the LISTDIRECTORY is simply one in HDFS, which implies solely the files gift within the highest level of a folder square measure browsed by the WebHDFS shopper. All the files within subfolders square measure thus unnoticed. This can be as a result of there's no algorithmic thanks to list filenames within the subfolders of a folder. Without the computer filename, this implementation of BlockHDFS cannot pass asking to the remainder API to calculate the hash and different needed parameters. Second, this NodeJs shopper, that is accountable for getting the info from WebHDFS and causating it to the Hyperledger material ledger, is about to execute periodically for a collection quantity of your time. Then, the Hyperledger creates a brand new quality once there's an amendment in information of file values like modification time, hash, and time interval. within the live production version, the NodeJs shopper ought to be created to figure in

realtime rather than capital punishment it sporadically by group action it directly into Hadoop, that may be a part of our future work.

CONCLUSION AND FUTURE WORK

Integrating blockchain with distributed file systems like HDFS will potentially improve security and traceability. Within the context of this paper, the original style of Hadoop is additionally optimized for file processes instead of security-by-design. Hence, during this paper, we tend to plan a brand new approach to introduce blockchain (and additionally, Hyperledger) to enhance the safety of the HDFS scheme. Within the current implementation, we've solely supplemental negligible data to the blockchain, however with BlockHDFS, one will simply add additional options suited to their application desires. For future work, BlockHDFS are often extended to work in period with the filing system and track all information between NameNode and DataNodes of the HDFS within the secure ledger with multiple nodes.

REFERENCES

[1] Apache hadoop, URL, http://hadoop.apache.org, 2006

[2] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: 2010 IEEE 26th Sym- Posium on Mass Storage Systems and Technologies (MSST); 3–7 May 2010; Incline Village, NV, USA, IEEE, Piscataway, NJ, USA, 2010, pp. 1–10.

[3] S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, C. Maltzahn, Ceph: a scalable, high performance dis- tributed file system, in: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06; 6–8 Nov 2006; Seattle, WA, USA, USENIX Association, Berkeley, CA, USA, 2006, pp. 307–320.

[4] F. Schmuck, R. Haskin, Gpfs: a shared-disk file system for large computing clusters, in: Proceedings of the 1st USENIX Conference on File and Storage Technologies, FAST '02; 28–30 Jan 2002; Monterey, CA, USA, USENIX Association, Berkeley, CA, USA, 2002.

[5] C. Ungureanu, B. Atkin, A. Aranya, et al., HydraFS: A high-throughput file system for the hydrastor content-addressable storage system, in: Proceedings of the 8th USENIX Conference on File and Storage Technologies, FAST'10; 23–26 Feb 2010; San Jose, CA, USA, USENIX Association, Berkeley, CA, USA, 2010, pp. 225–

[6] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.

[7] M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M.J. Franklin, et al., Apache spark: a unified engine for big data processing, Commun. ACM 59 (11) (2016) 56–65.

[8] Apache hive (2013). URL https://hive.apache.org/.

[9] M.N. Vora, Hadoop-hbase for large-scale data, in: Proceedings of 2011 International Conference on Computer Science and Network Technology; 24–26 Dec 2011; Harbin, China, IEEE, Piscataway, NJ, USA, 2011, pp. 601–605.

[10] T. Alladi, V. Chamola, R.M. Parizi, K.R. Choo, Blockchain applications for industry 4.0 and industrial iot: a review, IEEE Access 7(2019)176935–176951, https://doi.org/10.1109/ACCESS.2019.2956748.

[11] E. Nyaletey, R.M. Parizi, Q. Zhang, K.-K.R. Choo, Blockipfs—blockchain-enabled interplanetary file system for forensic and trusted data traceability, in: 2019 IEEE International Conference on Blockchain (IEEE Blockchain-2019); 14–17 Jul 2019; Atlanta, GA, USA, IEEE, Piscataway, NJ, USA, 2019, 00012.

[12] A. Yazdinejad, R.M. Parizi, A. Dehghantanha, K.-K.R. Choo, P4-to-blockchain: a secure blockchain-enabled packet parser for software defined networking, Comput. Secur. 88 (2020), 101629, https://doi.org/10.1016/j.cose.2019.101629.

[13] A. Yazdinejad, R.M. Parizi, A. Dehghantanha, K.R. Choo, Blockchain-enabled authentication handover with efficient privacy protection in sdn-based 5g networks, IEEE Trans. Netw. Sci. Eng. 8 (2) (2019) 1120–1132, https://doi.org/10.1109/TNSE.2019.2937481.

[14] A. Yazdinejad, R.M. Parizi, A. Dehghantanha, H. Karimipour, G. Srivastava, M. Aledhari, Enabling drones in the internet of things with decentralized blockchain-based security, IEEE Internet of Things J. 8 (8) (2021) 6406–6415.

[15] E. Androulaki, A. Barger, V. Bortnikov, et al., Hyperledger fabric: a distributed operating system for permis- sioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, EuroSys '18; 23–26 Apr 2018; Porto, Portugal, ACM, New York, NY, USA, 2018, pp. 1–15, 30:1–30:15.

[16] R.M. Parizi, Amritraj, A. Dehghantanha, Smart contract programming languages on blockchains: an empirical evaluation of usability and security, in: S. Chen, H. Wang, LJ. Zhang (Eds.), Blockchain–ICBC 2018, Springer, Cham, Switzerland, 2018, pp. 75–91.

[17] G. Wood, Ethereum: a Secure Decentralized Generalized Transaction Ledger. Ethereum project yellow paper 151, 2014, pp. 1–32 (2014).

[18] X.F. Zhang, Y.M. Wang, Research on intelligent medical big data systems based on hadoop and blockchain, 2021, EURASIP J. Wirel. Commun. Netw. (1) (2021) 7, https://doi.org/10.1186/s13638-020-01858-3.

[19] N. Abdullah, A. Hakansson, E. Moradian, Blockchain based approach to enhance big data authentication in distributed environment, in: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN); 4–

7 Jul 2017; Milan, Italy, IEEE, Piscataway, NJ, USA, 2017, pp. 887–892.

[20] M.S. Sahoo, P.K. Baruah, Hbasechaindb—a scalable blockchain framework on hadoop ecosystem, in: R. Yokota, W. Wu (Eds.), Supercomputing Frontiers, Springer, Cham, Switzerland, 2018, pp. 18–29.

[21] B.C. Neuman, T. Ts'o, Kerberos: an authentication service for computer networks, IEEE Commun. Mag. 32 (9) (1994) 33–38.

[22] P. Koopman, Embedded system security, Computer 37 (7) (2004) 95–97.

[23] Apache nifi, URL, https://nifi.apache.org/.

[24] P.P. Sharma, C.P. Navdeti, Securing big data hadoop: a review of security issues, threats and solution, Int. J. Comput. Sci. Inf. Technol. 5 (2) (2014) 2126–2131.

[25] Webhdfs rest api, URL, https://hadoop.apache.org/docs/stable/h hadoop-project-dist/hadoop-hdfs/WebHDFS. html.

[26] O. Kiselyov, A network file system over http: remote access and modification of files and files, in: USENIX Annual Technical Conference, FREENIX Track, 1999, pp. 75–80.

[27] Nmap. URL https://nmap.org/.

[28] Zenmap Introduction. URL https://nmap.org/zenmap/.

[29] M. H. Etzel, D. W. Faucher, D. N. Heer, D. P. Maher, R. J. Rance, Data encryption key management system, US Patent 6,577,734 (Jun. 10 2003).

[30] E. Androulaki, A. Barger, V. Bortnikov, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, EuroSys '18; 23–26 Apr 2018; Porto, Portugal, ACM, New York, NY, USA, 2018, pp. 1–15.

[31] T. Ylonen, SSH € –secure login connections over the internet, in: 6th USENIX Security Symposium, Focusing on Applications of Cryptography; 22–25 Jul 1996; San Jose, CA, USA vol. 37, 1996.

[32] X. Yu, Cyclic redundancy check computation algorithm, 802,038, US Patent 6 (Oct. 5 2004).

[33] B.J. Nelson, Remote Procedure Call. Ph.D Thesis, Carnegie Mellon University,Ann Arbor, MI, USA, 1981.