Journal of Big Data

RESEARCH Open Access

IoT Big Data provenance scheme using blockchain on Hadoop ecosystem

Houshyar Honar Pajooh[1*], Mohammed A. Rashid[1], Fakhrul Alam[1,2] and Serge Demidenko[1,2]

*Correspondence:
h.pajooh@massey.ac.nz
[1] Department of Mechanical & Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand
Full list of author information is available at the end of the article

Abstract

The diversity and sheer increase in the number of connected Internet of Things (IoT) devices have brought significant concerns associated with storing and protecting a large volume of IoT data. Storage volume requirements and computational costs are continuously rising in the conventional cloud-centric IoT structures. Besides, dependencies of the centralized server solution impose significant trust issues and make it vulnerable to security risks. In this paper, a layer-based distributed data storage design and implementation of a blockchain-enabled large-scale IoT system are proposed. It has been developed to mitigate the above-mentioned challenges by using the Hyperledger Fabric (HLF) platform for distributed ledger solutions. The need for a centralized server and a third-party auditor was eliminated by leveraging HLF peers performing transaction verifications and records audits in a big data system with the help of blockchain technology. The HLF blockchain facilitates storing the lightweight verification tags on the blockchain ledger. In contrast, the actual metadata are stored in the off-chain big data system to reduce the communication overheads and enhance data integrity. Additionally, a prototype has been implemented on embedded hardware showing the feasibility of deploying the proposed solution in IoT edge computing and big data ecosystems. Finally, experiments have been conducted to evaluate the performance of the proposed scheme in terms of its throughput, latency, communication, and computation costs. The obtained results have indicated the feasibility of the proposed solution to retrieve and store the provenance of large-scale IoT data within the Big Data ecosystem using the HLF blockchain. The experimental results show the throughput of about 600 transactions, 500 ms average response time, about 2–3% of the CPU consumption at the peer process and approximately 10–20% at the client node. The minimum latency remained below 1 s however, there is an increase in the maximum latency when the sending rate reached around 200 transactions per second (TPS).

Keywords: Internet of Things, Hyperledger fabric, Blockchain, Big Data, Data provenance, Hadoop, Traceability

Introduction

Over the past decades, data generated by the massive implementation and use of the *Internet of Things (IoT)* have been growing exponentially. The global Big Data market size has been projected to grow from USD 138.9 billion in 2020 to USD 229.4 billion by 2025 [1]. This unprecedented increase of data acquisition across many fields [2] (such

as healthcare, manufacturing, retail, logistics, transportation, etc.) allows for gaining meaningful in-depth insights. The extraction of meaningful insights from Big Data (e.g., volume, velocity, and representation) require a robust structure to facilitate the data storage, analysis, and processing in a secure, distributed, and scalable manner [3]. Big Data Analytics is an emerging field dealing with processing and analyzing vast data volumes [4]. The tremendous increase of data volumes within the Big Data ecosystems requires a robust solution to ensure information integrity so the correct knowledge can be derived from the analysis.

Blockchain offers a promising architecture for distributed large data storage and protection. A group of nodes and users within a blockchain network works cooperatively to structure the public ledger that contains the validated and recorded transactions as blocks. The data in IoT applications can be stored in off-chain storage (Big Data systems) while the data pointer to the off-chain storage can be kept in the blockchain system. When a data entity from the big data system is required, the blockchain accesses the specific data entity through a trusted environment. The user authentication is maintained by the distributed blockchain miners instead of a third-party auditor or a trusted centralized server. This study considers the decentralized storage for IoT data as off-chain Big Data system in a distributed manner while an entity can easily locate the address through the blockchain system. The third-party auditor and centralized trusted server are eliminated and the access to IoT data is managed by the blockchain nodes. The blockchain also manages the authentication of the users. The proposed work provides the accountability and tractability of IoT data where activities such as data modification and data access can be recorded in the blockchain.

Various approaches have been put forward to implement blockchain in several real-world applications. The Secured Map Reduce (SMR) is a security and privacy layer between HDFS and MR Layer (Map Reduce) introduced in [5]. The research work promotes data sharing for knowledge mining and address the scalability issues of privacy. The state-of-the-art security and privacy challenges in big data as applied to healthcare industry is reviewed in [6], The research work explores the security and privacy issues of big healthcare data and discussed ways in which they may be addressed. A permissioned blockchain is deployed for Halal supply chain to maintain secure transactions where the proposed model considers the transaction speed and rate to transfer data in effective manner [7].

Big Data analytics operating on cloud-based systems has been exploited widely. It has become the technology norm in extracting data-driven knowledge. The existing sensor-based IoT ecosystems (formed from integrating cloud-based Big Data analytics and wireless technologies) span a broad range of applications such as smart homes, smart cities, smart healthcare, etc. However, practical integration of IoT and Big Data systems face many issues such as security and privacy, non-interoperability, scalability, data traceability, and management. This hinders the true potential of such systems. Security concerns associated with data privacy, integrity, safety mechanisms, and quality could negatively affect Big Data systems applications [8]. The existing solutions fail to address the Big Data auditing challenges in cloud platforms efficiently. This could lead to security issues in computing-based Big Data storage. The multi-layer blockchain paves the way to address the privacy and security of IoT through a layer-based structure [9].

Local authentication and authorization are deployed to ensure the security of small IoT devices. *Hyperledger Fabric (HLF)* blockchain platform is a feasible approach to address the security and privacy challenges of edge computing devices within the IoT ecosystem. It can also provide the data provenance for generating data from IoT devices within the HLF and off-chain storage [10]. The details of implementing a layer-based blockchain model in an IoT environment including mathematical modelling and assumptions are described in our previous works [9, 10] along with the implementation of lightweight authentication mechanism for constrained IoT devices within the blockchain platform.

The most critical challenges and issues associated with various Big Data applications and techniques are security and privacy, infrastructure scalability, data interpretation, intelligence, real-time data processing, and data management. Among them, security and privacy are considered to be the most important [11]. The verification and integrity of user data within an untrustworthy infrastructure provided by a *cloud service provider (CSP)* is another critical challenge. Big Data characteristics consisting of variety, volume, and veracity raise concerns about efficient Big Data security mechanisms [12]. The aforementioned concerns and issues require investigating a robust mechanism that can verify the integrity of the outsourced data for Big Data storage in the cloud environment.

The majority of existing solutions incorporate *third-party auditor (TPA)* programs to maintain data integrity based on log files. This process increases the required storage size as well as communication and computation overheads. At the same time, it also brings many security concerns. Various solutions and practices have been explored to preserve data confidentiality and provide information security. In recent times the blockchain technology has received significant attention from many researchers as a promising solution to provide security and privacy in Big Data systems. The blockchain is defined as a number of nodes joined in a peer-to-peer manner maintained by the *distributed ledger technology (DLT)*. The study [13] considers the blockchain to enable efficient data collection and secure data sharing in a reliable and safe industrial IoT environment. The integration of blockchain with edge computing servers facilitates the security of the data collection process from IoT devices and the integrity of collected data [14]. Blockchain provides a robust structure for efficient and secure data collection in mobile ad-hoc networks [15]. Besides, the blockchain framework ensures data immutability, non-repudiation services, and network management capabilities. The decentralized architecture of the blockchain and its unique advantages make it a promising solution for securing big data services and protecting data privacy. Nonetheless, direct implementation of the blockchain technology on existing auditing systems is not practical [16]. The performance of blockchain system is degraded with the increase of the volume of data stored in the ledger. Accordingly, the deployment of the centralized auditing program into the decentralized blockchain network is challenging.

IoT systems face challenges in performing various identity managementfunctions, maintenance of the trustworthiness of data, access control to data within the network, and detection of abnormal behaviours. Data provenance is a solution to tackle these challenges. It includes recording information about data operations, and data origins as well as analyzing the data history from their source to the current state. Blockchain offers a distributed data storage. It can be deployed to provide data provenance for various applications by recording data operations from blockchain transactions. Embedding

the data provenance (enriched by blockchain technology) into Big Data applications enhances system security and privacy while ensures data availability. The blockchain-enabled data provenance mechanism for Big Data applications in IoT systems guarantees data verifiability and integrity. This is because the data operations are recorded in the form of the transaction by every block in the blockchain network. Different devices within the IoT edge cloud architecture impose various trust concerns on the systems. Hence, a provenance mechanism is applicable to record the origin of multiple sensor data to meet these concerns [17]. Nonetheless, the blockchain-based provenance system scalability can be enhanced by integrating the high capacity of Big Data systems such as the *Hadoop Distributed File System (HDFS)*. Smart contracts combined with cryptographic methods maintain the task automation within the blockchain network. The integration of smart contacts helps to build up a secure environment for the IoT Big Data applications through a comprehensive data provenance management system. This study aims to provide the data provenance, integrity, traceability, and accountability for a large volume of data generated by a very large number of IoT devices and stored in a secure and verifiable Big Data ecosystem.

Blockchain technology paves the way to provide security and privacy for large-scale IoT data storage as well as to enhance the decentralized storage application, eliminate the centralized trust server, facilitate data traceability and accountability. Although many research efforts attempted to address the security and privacy challenges of Big Data systems, the authors are not aware of any studies on the application of blockchain technology that would comprehensively address the data traceability and data provenance for Big Data systems on large-scale IoT environments. The main goal of this paper is to address the outlined research gaps by implementing the HLF blockchain framework to maintain data provenance and auditing on Big Data systems within the large-scale IoT network without third-party auditing interventions. HLF blockchain is deployed to enhance data security by implementing mutual authentication and overcoming communication and computation overheads. In summary, the main contributions of this work are as follows.

1) HLF blockchain scheme is developed to provide secure data storage for Big Data systems in a large-scale IoT network. The proposed model maintains data privacy preservation, ensures a secure connection to a Big Data system through the HLF network, and guarantees data collection security. The centralized trust server is eliminated through implementing the HLF blockchain technology.
2) A two-layer security framework is proposed that involves HLF blockchain and a Big Data system. Trusted entities are linked to HLF, and third-party auditing parties are eliminated to reduce the compromised auditor's risk. The network scalability is enhanced by incorporating edge computing to maintain IoT data computation as well as to collect and forward data to the blockchain and off-chain storage.
3) A model is proposed to store the lightweight verification checksums and data pointers in the blockchain ledger to reduce the communication and computation overheads. The HLF blockchain performs data provenance while the actual metadata are stored in off-chain storage after being verified by the blockchain. Extensive experiments were conducted through a prototype implementation on a Hadoop system to

evaluate the performance of the proposed scheme in terms of throughput, response time, latency, communication, and computation cost.

The rest of the paper is organized as follows. "Background" introduces blockchain technology, security settings, Big Data systems, and the primary settings of the model. An overview of relevant state-of-the-art literature sources for the different data provenance solution approaches is presented in "Related works" section. In "System model and architecture" section, the proposed model is extended to protect large-scale IoT data storage. The system implementation is presented in "System implementation" section. Detailed model analysis and performance evaluations are presented in "Results and discussions" section. Finally, the findings are summarized in "Conclusion" section. along with outlining potential future research directions.

Background

Blockchain and Big Data systems are the two main components of the blockchain-enabled IoT data provenance framework. Blockchain provides a security and privacy basis. It guarantees the authorization and authentication for data owners and users with specific access and allows them to perform data analysis. Meanwhile, blockchain records storing the lightweight verification tags on the blockchain ledger to maintain the verifiability, integrity, and traceability of data are stored in off-chain storage. An overview of the technologies used in the framework is presented below.

Blockchain

Blockchain, as an open-source digital distributed ledger, is one of the most prevalent innovations broadly deployed in various areas [18]. Nodes within the distributed blockchain network communicate in a *peer-to-peer (P2P)* manner while the need for a centralized authority is eliminated. Blocks are the list of records wherein stored information is encrypted. All transactions are cryptographically marked and verified by all other participants holding replicas of the entire ledger and records. Thus, all records are immutable, tamperproof, synchronized, and cannot be changed when stored in the blockchain. Blockchain platforms can be categories into three types: *private*, *public*, and *consortium*. Public or permissionless blockchains such as Bitcoin and Ethereum [19] allow all entities to join the network without restrictions while anonymous participants can perform the verification process. On the contrary, participants are required to get permission to join the private blockchain or permissioned blockchain network while the blockchain is limited to the authorized participants belonging to an organization or group of organizations. Only selected nodes within the blockchain consortium can perform the verification process (Hyperledger Fabric [20] and Ripple [21]). In consortium blockchains, a specific group of nodes has access to the public ledger. The blockchain architecture is partially decentralized. Here, the consensus process is maintained by all participants based on specific rules. The key features of blockchain are as follows.

1) *Immutable* Blockchain (with its permanent and unalterable characteristics) provides an immutable framework where each node has a copy of the ledger. Transactions are

verified and validated by nodes before adding to the ledger. Participants are not able to make alterations to the data stored in the blockchain ledger.

2) *Distributed* The deployed standard protocols in blockchain facilitate orchestrating blocks build upon the group of transactions verified by participants based on a predefined set of rules. Besides, blockchain can synchronize and distribute the data among multiple networks.
3) *Decentralized* The architecture of blockchain eliminates the need for a central authority. The governance is done by the group of nodes maintaining the ledger. Network participants hold a copy of all transactions and record their replicated data using private keys. Thus, the risk of single-point failure vulnerability is eliminated.
4) *Consensual* Data consistency within the blockchain framework is maintained by the associated consensus algorithms, and the blockchain operation relies on that. The consensus process decides to select a group of active nodes and remove the false and corrupted transactions added to the ledger. Maintaining transaction data integrity is achieved when all nodes agree by executing consensus algorithms.
5) *Anonymous* All users communicate in the blockchain network in a P2P fashion. Users' identities cannot be disclosed while the encoded transaction details are visible to all participants.
6) *Secure* High degrees of security are provided by the immutable and decentralized blockchain through deploying various cryptography techniques. Each set of data is uniquely identified by implementing hash functions on information and robust firewall algorithms to protect the framework against unauthorized access. The data are tamperproof as each block in the ledger holds its associated hash information and the previous block hash data.
7) *Traceable* The blockchain transactions are digitally signed and time-stamped thus facilitating data traceability and auditability. Every block is permanently connected to its previous block enabling the data owner to trace the data within the blockchain framework.

Big Data systems

Big Data is a definition for large data sets that traditional data processing systems cannot efficiently interpret, collect, process, and manage using conventional methods and mechanisms. Big Data typically have the 4-V attributes, consisting of *volume*, *velocity*, *variety*, and *veracity* [22]. Many challenges are associated with data volume processing, such as modularity, imbalance, dimensionality, data nonlinearity, bias and variance, and computing availability. Variety indicates the collected data types, which are naturally heterogeneous and involved structured data, unstructured data, multi-structured data, and semi-structured data. The velocity presents the data generation speed while the veracity indicates the quality of data generated from various sources.

Big Data analytical systems facilitate knowledge extraction from multiple datasets for various purposes. The extracted information can be used in many applications, including smart cities, smart grids, e-health, logistics, transportations, mobile and wireless communications. The most popular data analytics frameworks in the industry are Hadoop [23, 24], MongoDB [25], Spark [26], and Strom [27].

Hadoop ecosystem

Hadoop is a framework to manage an orchestration of a cluster of computers with distributed processing based on the *MapReduce* programming model. Two main components of the Hadoop system are *MapReduce* (for parallel and distributed processing) and *Hadoop Distributed File System (HDFS)* (as storage of data in a distributed file system). Nodes within the Hadoop architecture are classified into *Master* and *Slave* ones. The master node performs the data collection and maps them to the respective slave nodes. The slave nodes maintain read/write operations in the file system and carry out the block creation, block deletion, and replication based on the name node rules. The master node then records all the operation results. The final results are formed by combining the sub results. The MapReduce determines the master nod as a job tracker and the slave nodes as task trackers. Therefore, the job scheduling, subtask distribution, and fault tolerance associated functions take place in the master node.

HDFS helps to address the storage challenges of Big Data sets by distributing the enormous volume of data among various computing resources and machines called the Hadoop cluster. The main components of the HDFS architecture are the *Name Node* (referred to as a master node), the *Data Nodes* (slave nodes), and the *Secondary Name Node* (the name node backup). The HDFS distributed system architecture is illustrated in Fig. 1.

MapReduce programming framework is implemented within the Hadoop system to perform the Big Data processing. The framework maintains the processing of large data sets in a parallel and distributed manner across the Hadoop cluster architecture. Map phase and Reduce phase are the two distinct tasks of the MapReduce process. The data process is happening in all machines with the Hadoop cluster. It is known as the Map phase. Combining the outcomes and forming the final results are referred to as the Reduce phase. MapReduce is one of the core pieces of Hadoop that performs big data analytics along with HDFS and *Yet Another Resource Negotiator (YARN)*. YARN is a technology sitting under the hood to manage all the resources of the cluster and to

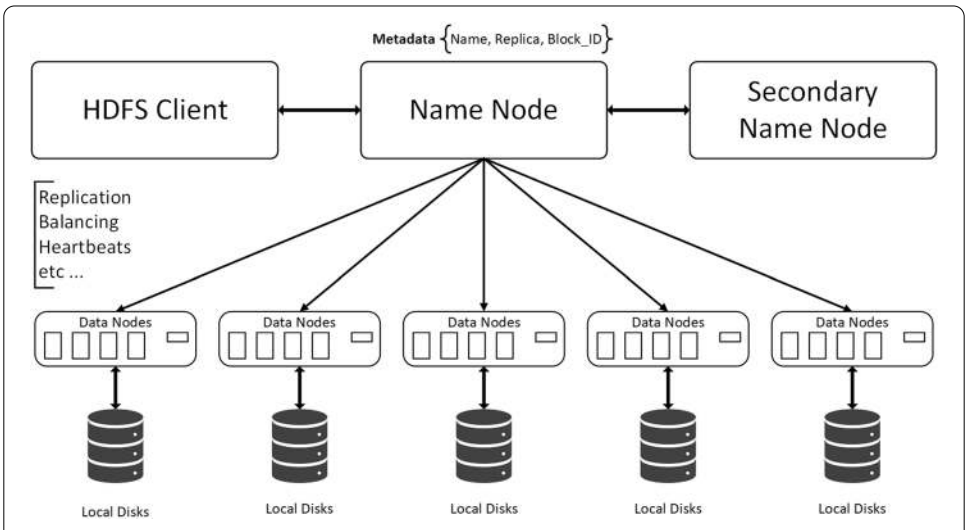

Fig. 1 The HDFS system architecture

assign computational resources for application execution. *NodeManager*, *AppManager*, and *Container* are the components of YARN. Figure 2 presents the detailed operations and system structure of MapReduce.

Related works

Data provenance and data integrity have been considered as critical elements of the security requirements for big data analytics in IoT-based solutions. They enable users to check the integrity of stored data in outsourced storage. Data provenance enhanced with blockchain technology is a promising solution to provide the trustworthiness of stored data through immutable and tamperproof information about the data origin and history of data records.

Blockchain-based data provenance in IoT

AgriBlockIoT [28] is a fully decentralized blockchain-based model to maintain the data traceability for Agri-Food supply chains. The mechanism provides immutable, fault-tolerance, and auditable records of the whole supply chain system from production to consumption. The history of the purchased product is recorded in the blockchain system thus enabling effective data retrieval for consumers. The proposed system collects data provenance including the data origins and operations performed on the data. The trust concerns coming from various IoT edge devices in cloud infrastructure are addressed by a provenance mechanism to record sensor data and origins of the related entities [17]. The provenance system structure is based on a combination of IoT edge devices organized with a blockchain network. Blockchain transactions are used to record all actions within the ledger with data provenance. *Physical Unclonable Functions (PUFs)* are utilized in BlockPro [29] to facilitate the data provenance and data integrity to achieve secure IoT environments with the help of the Ethereum blockchain and smart contracts.

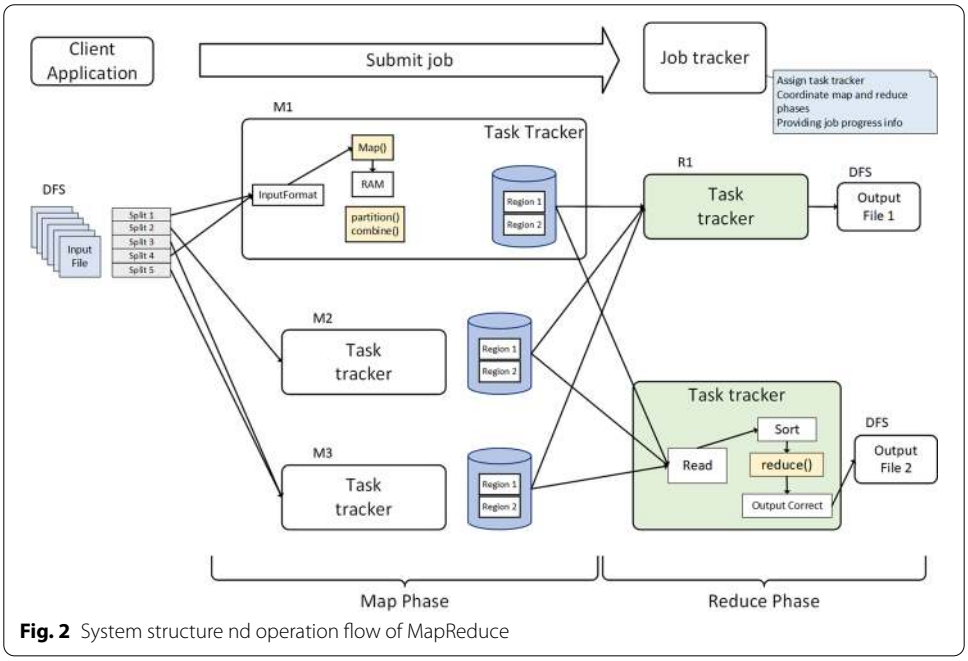

Fig. 2 System structure nd operation flow of MapReduce

PUFs produce unique hardware fingerprints for each device and deploy them to find data provenance and identify the data source.

A distributed database based on the blockchain was designed to guarantee data verifiability and integrity called ProvChain [30]. The Ethereum blockchain and two smart contracts are applied to maintain a decentralized digital ledger to ensure data integrity and prevent data tampering attacks. Data operations are stored in the local ledger while the blockchain records the provenance entry. The provenance retrieval from the blockchain network is maintained by a *Provenance Auditor (PA)* that keeps the local database. Research [31] reports an extensible and secure IoT data provenance framework based on a layered architecture consisting of smart contracts and Etherum blockchain implemented for a wide range of IoT applications. Shreya Khatal et al. [32] propose decentralized storage called *Fileshare* for file sharing within the secure environment based on blockchain to ensure the integrity and ownership of shared files. The introduced *Decentralized Application (DApp)* is built upon the Ethereum blockchain framework while smart contracts utilize a distributed file system in the data layer.

Blockchain-based data verification

Blockchain technology has recently attracted many researchers in various fields, including cloud data storage and data integrity, application of edge and fog computations, provenance, etc. [33]. The foundation of cloud storage systems is formed based on data storage. The challenges of storing cloud data securely are being investigated in [34, 35] and addressed by deploying blockchain techniques. Restructuring the history of data associated with each data operation or scientific result is a critical data provenance element. Nonetheless, it facilitates data management more efficiently in different applications such as scientific data and high-quality web data management. Works [36, 37] investigate embedding the data provenance mechanism into blockchain transactions to address the collection and verification issues.

Computing services with low latency and higher bandwidth are in place by applying the new edge and fog computing techniques with shared resources. Edge and fog computing security can be further enhanced by integrating the emerging blockchain technology to establish a trusted decentralized environment. Blockchain technology has been considered to ensure the privacy-preserving for applications on edge platforms. Besides, data storage and resource allocation applications are deployed using blockchain at the edge and fog computing level while the architectural security is further improved [38–41]. Although the existing works attempt to enhance the data provenance mechanisms and edge computing applications by replacing some functionalities with blockchain technology, they still rely on centralized entities with significant limitations and additional overheads generated from deploying the centralized entities.

Research [42] proposes a framework for data integrity based on a blockchain for peer-to-peer (P2P) cloud storage. Data integrity verification is deployed using rational sampling approaches to establish sampling verification effectively. A fixed third-party auditor is deployed to perform the integrity verification of operation logs based on blockchain in the cloud [43]. This method brings third-party auditor security drawbacks into the system while the computing and communication overhead being quite considerable. A certificate-less public verification scheme against procrastinating auditors with the aid

of blockchain technology is proposed in [44]. The main idea is based on recording each verification by auditors in the form of blockchain transactions. Moreover, certificate-less cryptography is deployed in the scheme to overcome certificate management issues.

Implementing blockchain in a distributed large scale IoT environment with Big data storage providing the protection is a challenging task. The most significant issue is providing a light-weight authentication mechanism to manage the identities of users and IoT devices through a blockchain system. Most of the research works consider authentication and other security primitives in a centralized server. Besides, providing a secure channel where data provenance and accountability can be maintained without intervention from third party and trusted central server is a major limitation of the previous works. In our work, we take advantage of light-weight authentication model to achieve effective and efficient authentications for the users and IoT identities. The works sited in the literature review suffer from scalability problem that has been addressed in our proposed scheme with multi-layer blockchain approach extending to Hadoop database as off-chain storage of the underlying database. Hadoop is a distributed and scalable Big Data storage and supports random, real-time read/write access to Big Data. Furthermore, the majority of the previous studies consider the Cloud-IoT environment in which a large number of users and devices share data through the cloud computing infrastructure. However, the cloud can not provide a scalable platform and suffers from a lack of supporting a vast number of users. Therefore, the existing research works are limited to a certain number of devices and users. These challenges have been addressed in our proposed framework. In addition, our work considers processing massive data in IoT devices through lightweight algorithms to overcome the limitation of energy efficiency and processing performance of the current approaches.

System model and architecture

Blockchain technology and Big Data integration have been considered as potential solutions to address large-scale real-world problems. The exponential growth in the generated data presents its own security and privacy challenges and issues associated with data sources reliability and data sharing. The challenges of the Big Data ecosystem can be answered using unique features of blockchain technology such as decentralized storage, transparency, immutability, and consensus mechanisms. The integration of them can further enhance Big Data security and privacy, improve data integrity, provide fraud prevention, facilitate real-time data analytics, expand data sharing, enhance data quality, and streamline data access.

This work aims to develop a blockchain-enabled public data provenance and auditing model in the Big Data ecosystem (Hadoop ecosystem) to provide a more efficient and secure framework than the reported solutions. Blockchain offers a decentralized database that records the history of all transactions appended to the shared ledger and enhances data traceability. The information inside the Big Data ecosystem in many applications is shared with multiple workers and writers, while most of them may be non-trusting participants. Blockchain as a resilient framework is a feasible solution to eliminate a third-party intermediary, provide automated interactions among multiple transactions in the shared database, and enhance auditability. To achieve the goal of the research, a distributed provenance tracking architecture is designed while deploying an

application built on top of the HLF and Hadoop ecosystem. The proposed data provenance model aims to identify the way the data was derived and to provide data *confidentiality*, *integrity*, and *availability*. The HLF permissioned blockchain having registered members offers the above-required functionalities.

The blockchain-based high-level scheme

The architecture of the system includes three layers: a blockchain layer, a Big Data system (off-chain storage) layer, and an authentication provider layer. The first component is the HLF network implemented and running in *Docker containers* [45] and associated client libraries for multiple interactions with the HLF. The second component is the Big Data system (Hadoop ecosystem, in this case) operating as off-chain storage. The communication with other parts of the system is managed through a client library to initiate and perform multiple tasks and operations. The proposed model aims to provide seamless records of provenance data in a tamperproof and immutable blockchain ledger, ensuring data storage and access to a pluggable Big Data storage service. HLF blockchain framework is deployed to record all provenance data entries. Multiple data operations can be stored within each block in the system. Data operations, as well as invoke and data querying, are recorded in the blockchain ledger. The identity of devices needs to be registered and stored in the shared ledger before running the HLF blockchain. Each gateway collects data from connected devices and sends them to a higher level for verification and storage. The registration request is sent to the gateways (IoT applications to edge IoT nodes). It includes the required information such as device ID, gateway identity, and timestamp. The gateway runs the *ChainCode (smart contract)* in the local blockchain to perform the device registration. The mutual authentication model is designed and implemented to provide the device's authentication before joining the network to ensure a secure and trusted environment. Secure communications between entities can be then established through the implemented blockchain network.

Figure 3 demonstrates the blockchain-based data provenance system model. It verifies data integrity by finding the location of the data item and associated checksum. The lineage of data for new items is accessible via storing the references to the data items deployed to create it. Data operations are recorded to have good visibility on the time and the clients who stored the data items or manipulated the data object. The records are maintained based on the certificate ID used to invoke the transaction. Such a design provides a data provenance framework based on the HLF blockchain offering data security, privacy, and auditability for Big Data systems.

Hyperledger framework

Multiple HLF processes are orchestrated and configured to be run on different nodes using Docker containers. Nodes in the HLF network run a peer process and maintain the shared ledger through various transaction proposals. The client library initiates the transaction proposal using the HLF software development kit functions, which are cryptographically signed with a certificate generated by the *Certificate Authority (CA)*. The critical element in the HLF framework is the peer process. It holds a replica of the shared ledger by running the ChainCode. Running more peers helps

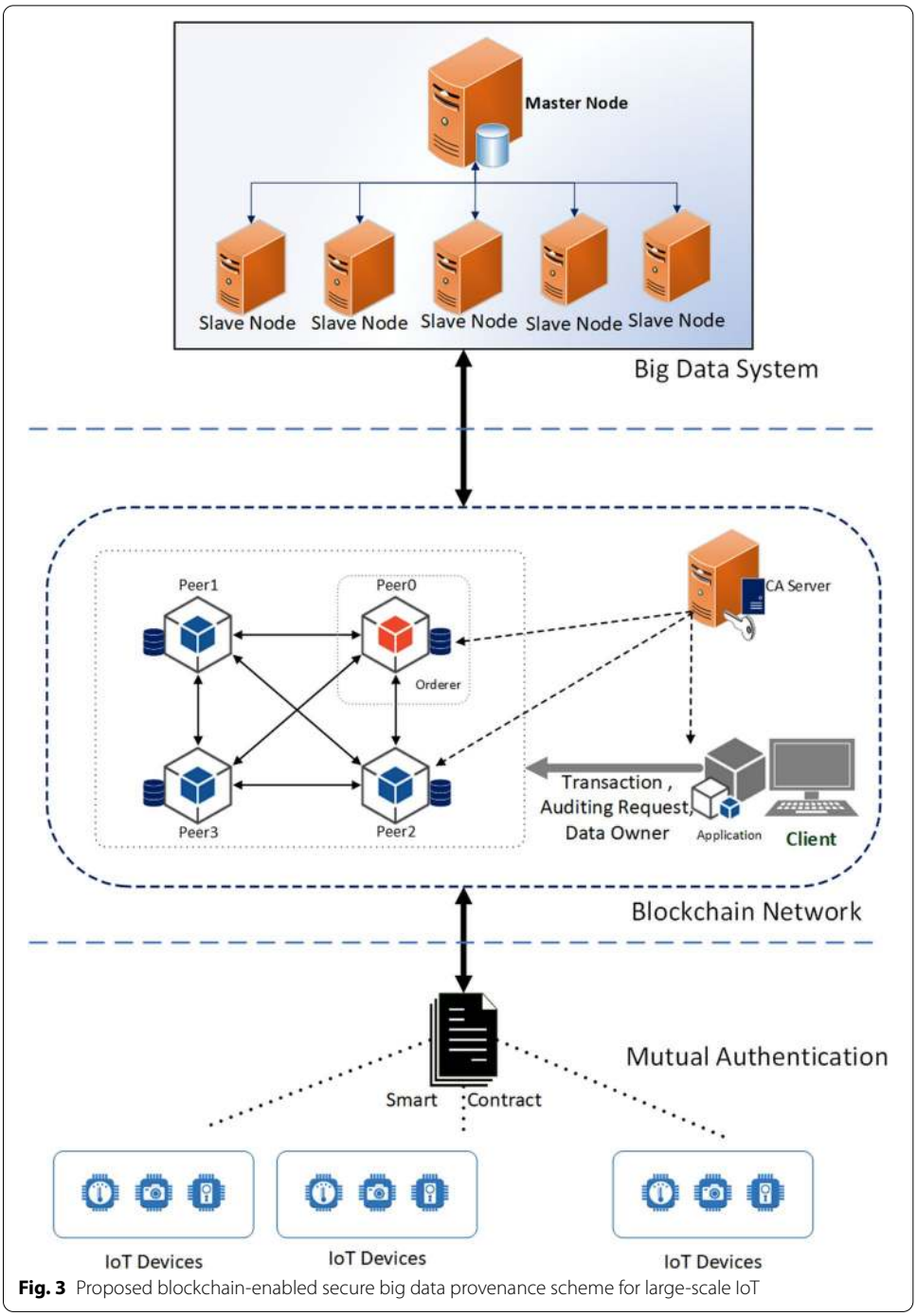

Fig. 3 Proposed blockchain-enabled secure big data provenance scheme for large-scale IoT

to achieve higher performance. At the same time, one peer per organization is sufficient to run the HLF network. The ordering service handles the block ordering (based on deterministic consensus protocol) and validates the blocks that peer processes have proposed. The single-orderer architecture is considered in the reported model using the built-in HLF RAFT consensus algorithm. Figure 3 presents the proposed

blockchain-enabled secure Big Data provenance scheme for a large-scale IoT, including the HLF framework components.

Hadoop data storage

The distributed shared ledger implemented through the HLF blockchain has limitations in terms of data storage. The HLF performance degrades with the growth in the ledger's size resulting from increasing the blockchain platform's shared ledger. The proposed model stores the provenance of data in the shared ledger (a small portion of the metadata). The actual metadata is placed in the Hadoop ecosystem to tackle the issues mentioned earlier. In this way, the data is stored in the off-chain storage. The data checksums are computed to perform the data verification and integrity checks. Hence, the HLF blockchain can verify stored data integrity by comparing the immutable recorded information in the shared ledger with the checksum of stored data in the Hadoop system. A ChainCode is developed to facilitate these operations running in each peer node within the HLF network. The built-in client library sends the data checksum and provenance data. The file-store operators are not needed and thus are eliminated. A flexible distributed Hadoop ecosystem is introduced as a pluggable storage solution to accommodate secure and verified data. The client facilitates the data invocation process by putting the data in the Hadoop storage and sending information to the blockchain for verification. The data query operations initiate the ledger side to acquire the location and address. Then the data is retrieved from the Hadoop storage.

ChainCode

The ChainCode runs on the peer nodes, maintains the data query, and appends data on the information stored in the shared ledger. It is the main component in the model with several functionalities to automate the tasks within the blockchain platform. All peer nodes have access to the functionalities implemented in the ChainCode. Storing and retrieving data from the shared ledger are automated by the ChainCode. The proposed design considers storing checksums of all data objects, data addresses and locations, information about workers who stored the data, information on creating an object, data lineage, timestamp, certificate ID, and additional fields that can be customized for various data structures (e.g., JSON structure). The process starts with the ChainCode functions (invoked as parameters associated with the data) to begin storing data in the HLF ledger. A specific function is designed in the ChainCode to perform the data retrieving functionalities. The data can be queried based on the data items assigned to a particular stored key and the data iterations. The query of data collections is provided through various query definitions within the ChainCode, either by key range or the iteration history.

Client library

The client library is developed and built using the *Software Development Kit (SDK)* [46] to interact with the HLF blockchain platform for data verification and provenance operations. The client library is a core element operating as a middleware for all other applications that need to interact with blockchain and store data or record the provenance data. The client application communicates with both blockchain and Hadoop systems for various operations as different distributed workers. It can be integrated into the peer

node or work as a separate node within the HLF blockchain system. Several client nodes with their associated client applications can run the HLF blockchain as an overlay network in the background and perform various tasks relevant to Big Data analytics. The client applications control the fraction of data stored in the HLF shared ledger and the metadata that need to be stored in the Big Data system.

Edge computing

The edge computing device is a central node to implement the blockchain-based IoT Big Data storage scheme. It offloads the tasks from small IoT devices and maintains significant energy savings. Besides, it performs the associated computations, manages data storage, and relays transactions and messages for IoT devices. The edge computing node contains the IoT device identification and authentication information. It stores the identification information of all interconnected IoT devices and provides a pair of keys for each device to perform the authentication through implementing a lightweight mutual authentication protocol.

The authentication procedure is shown in Fig. 4 for each IoT device to the edge computing server. The generated messages and transactions by IoT devices are managed and created by the IoT edge node. The HLF blockchain framework runs on the edge computing nodes, and the edge server conducts signing valid transactions, including the IoT device signature. The sensitive data are then verified to be ready for storage in the Hadoop ecosystem while the data checksums and related operation tags are stored in the HLF blockchain. The edge servers collect all verified and trusted data and send them to the Hadoop distributed file system. The collection of data from interconnected IoT

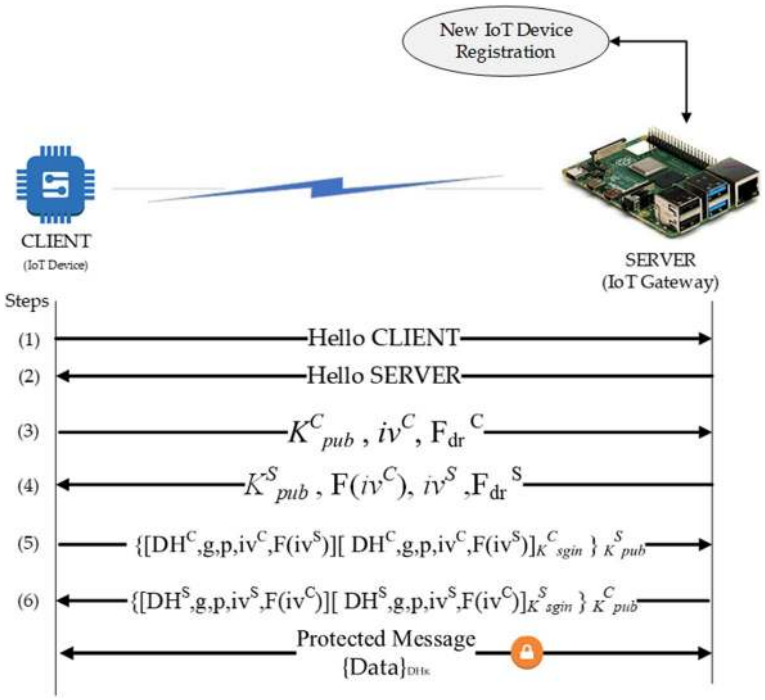

Fig. 4 Local mutual authentication procedure at IoT edge computing [10]

devices is a continuous process. The locations and addresses of the stored data are determined in the HLF blockchain for further verification and traceability operations. The details of the authentication and authorization process and the procedures to implement it within the layer-based structure are fully covered in the earlier research [10].

System implementation

The interaction with the ledger in HLF is possible through executing the defined ChainCode. The ChainCode is responsible for storing the data provenance and handling various data queries. Hence, the system implementation starts with defining specific ChainCode operations consisting of storing data provenance, querying the lineage of data, and retrieving data lineage. The IoT applications require a lightweight ChainCode to be implemented on endorsing peers to address the limitation of IoT devices in terms of their communication capacity, storage, and processing power.The access of the ChainCode to external resources is limited to ensure that the ChainCode can provide the same results for all endorsers. The ChainCode is designed to support different operations associated with data provenance and traceability of data within the ledger and the attached off-chain storage. The ChainCode specific operations in the proposed system include storing the data provenance related to an item, querying item checksums, retrieving an object with the associated transaction ID, extracting the version of an object based on its transaction ID, retrieving the lineage of the data item, retrieving the history of a data objects, querying the key-range of the list of items, retrieving the provenance information, and providing a specific version of an object and the related transaction ID. The main implementation concern is to make the ChainCode lightweight that can address the limitations of IoT devices and allocate a significant part of functionalities to the client applications. The implemented system consists of distributed peer nodes that are at the centre of communications between network elements and the off-chain storage (Big Data ecosystem).

The performance of the proposed model was evaluated for the system throughput, response time, latency, and resource consumption (memory, CPU, network) metrics. The evaluation was further expanded to cover the scalability of the distributed large-scale IoT network environment. The measurements were conducted by implementing a benchmark application on top of the *node package manager (NPM)* libraries run on the client node. Besides, various Linux-based tools and utilities were deployed to monitor the system's performance. To emulate a large number of IoT devices, the customized *Locust* [47] was deployed on an independent server interconnected with the edge computing devices in the same LAN. The experiments were conducted by emulating 100–2000 IoT devices connected directly to the edge IoT servers to send messages and transactions. A maximum number of 500 IoT devices was considered to be managed by each edge device. The edge server stored the identification of all connected IoT devices and authenticated them within a trusted HLF environment by implementing a mutual authentication scheme described in "Edge computing" section.

The performance analysis was carried out for the proposed model for various workloads and environment parameters. Moreover, a diverse set of interaction performances was observed to explore the improvement or degradation caused by different

parameters and configurations of the model. Several benchmarking applications build on NPM libraries run on client nodes were employed to perform the benchmarking processes.

Often, stakeholders need to find out which benchmarking model is suitable for their applications and particular use cases since different methods differ in terms of involved parameters and phases. To address this challenge, the HLF performance guidelines and HLF performance metrics documented in the Hyperledger Performance and Scale Working Group white paper [48] were considered to conduct the benchmarking of HLF V1.4. Real-time data reporting was deployed and statistic data on resource utilization were collected and monitored.

Experiment setup

The setup of the developed system prototype consists of five units of ARM-based *Raspberry Pi (RPi)* 4B, a client-server, and a Hadoop system as off-chain storage. The hardware and software specifications of RPis are summarized in Table 1. The RPis, client-server, and Big Data ecosystems were interconnected within the same LAN. The peer docker containers run on each RPi, and one node was assigned as the orderer node. Unofficial docker images of HLF version 1.4 were modified and established on each RPi device. The docker images were compiled to suit the ARM64 architecture of the RPi. Performance measurements were conducted by a client desktop computer using a client application build on the NPM libraries. The client application was developed using HLF node SDK. The ordering type (using the solo type of order in HLF) indicated that the consensus was achievable by a single ordering node implementing a sorting algorithm. New block generation was done based on specific parameters that have been defined in the client application.

The Hadoop cluster was configured with one master node and five slave nodes. The cluster was equipped with 48 CPU core and 35 TB local storage. The details of the Hadoop cluster configuration and associated software are presented in Table 2. The cluster had dedicated switches. It worked in the same networking structure. The same as mentioned in "Blockchain-based data verification" section, Yarn maintained the resource management. It facilitated resource monitoring for active nodes including the job details and correspondent histories. The HDFS was configured in the master node (name node), secondary name node, and five worker nodes (data nodes). Figure 3 shows the proposed model and the system under test architecture used for the performance measurements.

Table 1 Raspberry hardware specifications

Type of device	CPU cores	Memory
Raspberry Pi Computer Model B	Broadcom BCM2711 Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	4GB LPDDR4
Client node	Intel (R) core(TM) i-7-6700 CPU @3.4 GHz	8 GB

Table 2 Hadoop cluster experimental setup and specifications

Node configuration	Hardware	Specifications
Server configuration	Processor	2.9 GHz
	Main memory	64 GB
	Local storage	10 TB
Node configuration	CPU	Intel® Xenon® CPU E3-1231 v3@ 3.40 GHz
	Main memory	32 GB
	Number of nodes	5
	Local storage	6 TB each, 30 TB Total
	CPU cores	8 each, 40 total
Software	Operating System	Ubuntu 16.04.2 (GNU/Linux 4.13.0-37-generic x86 64)
	JDK	1.7.0
	Hadoop	2.4.0
	Spark	2.1.0
Workload	Varying data sizes and batch sizes	Submitted by the HLF client application

Results and discussions

This section presents the results attained after implementing the proposed model and sending various batch sizes and workloads to evaluate the performance of the entire architecture. We have assessed the performance of the proposed model by measuring multiple parameters consisting of the system throughput, response time, latency, and resource consumption (memory, CPU, network) metrics. Each measurement was conducted with some repeats, and the average obtained results were plotted in each graph.

Throughput and response time measurements

The benchmark application was developed on top of the client library to generate transaction batches to the network. A timer was associated with each transaction. In addition, the timer was allocated for every set of transactions. The benchmark application calculated the response time of a transaction and the total average time while considering the number of successful and failed transactions for various data set and batch sizes. The benchmark application was powered with the ability to store the data in the HLF ledger or the Hadoop system. The performance evaluation was initiated by several transactions submitted together. The results indicated that the throughputs and response times were affected by the size of the data. However, the impact was not significant if the data provenance and the transaction tags were only stored in the blockchain ledger. As stated earlier, the provenance of data was stored in the blockchain ledger, and the actual metadata were placed in the Hadoop ecosystem off-chain storage.

The performance was affected when the Hadoop system was involved in storing the metadata (since the client application needed to consider the time for calculation of data checksums, operation tags, and the time to store the data in the Hadoop system). Figure 5 illustrates a degradation in the performance with the growth in the bath sizes.

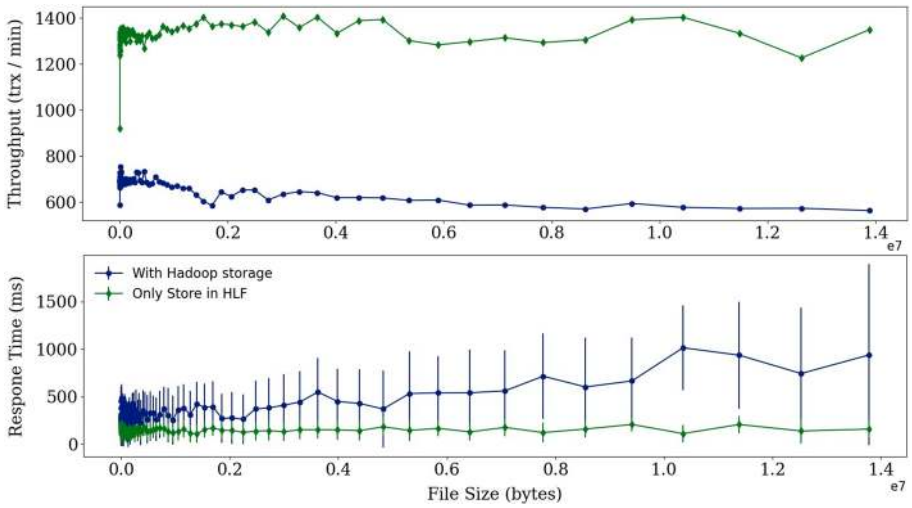

Fig. 5 Throughput and response time for various batch sizes

The obtained results (600 transactions per minute and 500 ms average response time) can be considered as very promising according to the HLF performance guidelines and HLF performance metrics documented in the *Hyperledger Performance and Scale Working Group* white paper [48]. One of the main limitations came from the client node's hardware capabilities and the peer process hardware constraints. The HLF employed the *Execute-Order-Validate* and Commit transaction model. Therefore, the system needed to perform the required operations for each data object, resulting in degradation in the throughput and increased response time. Besides, the system needed to consider the time for storing the data provenance in the HLF ledger, calculating the checksum of data objects, and storing the metadata in the Hadoop system.

To address the challenge,the network was made to include multiple clients. More endorsers were required to improve the overall throughput and response time performance. With the small number of transactions, the throughput was slightly lower, while the increase in the number of transactions led to some rise in the throughput. At the same time, it could be noted that the throughput was approximately constant for a certain number of transactions.

The latency measurements were done by running multiple rounds of the benchmark to submit various transactions with different sending rates(from 10 *Transactions-per-Second (TPS)* to 500 TPS) for different block sizes. The goal was to measure the maximum, average, and minimum transaction latency.

The results indicate that during the experiments, the minimum latency remained below 1 s. However, there was an increase in the maximum latency when the sending rate reached around 200 TPS. This was due to the rise in the number of ordered transactions waiting in the verification process queue during the validation phase that significantly increased the commit latency.

Since the system setup deployed a solo-orderer configuration, other orderer types needed to be employed along with different configurations. Consequently, the validation phase was considered as being a bottleneck in the overall system performance.

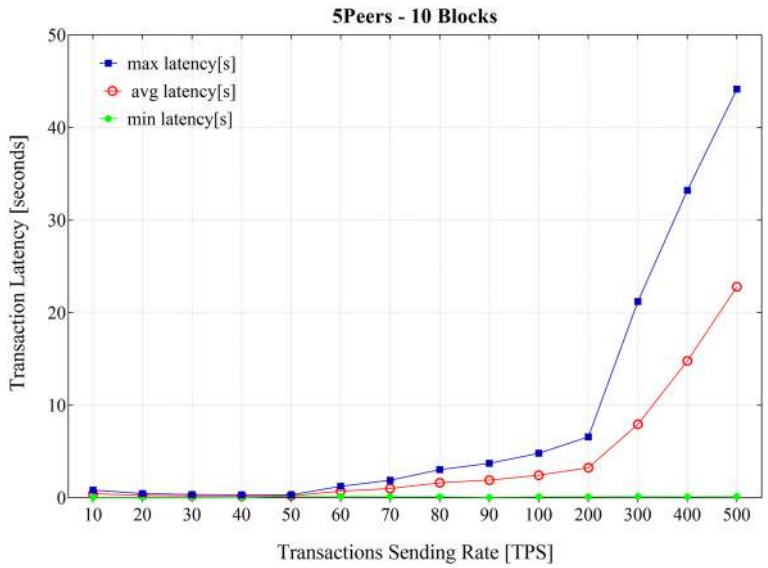

Fig. 6 Latency vs. transaction sending rate (5 Peers 10 Blocks)

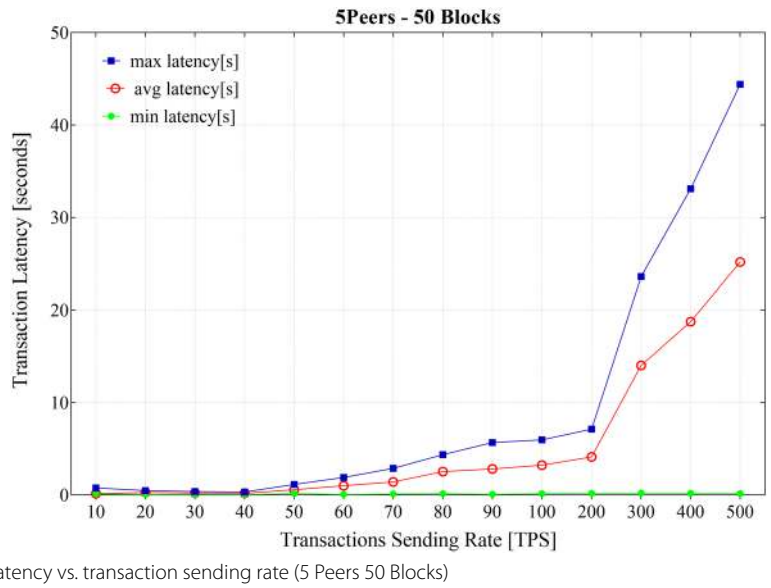

Fig. 7 Latency vs. transaction sending rate (5 Peers 50 Blocks)

Hence, there was a need to deploy a smaller block size with a low transaction rate for IoT applications to have lower transaction latency. In contrast, higher transaction rates needed larger block sizes to achieve higher throughput and lower transaction latency (the results of latency measurements for various block and batch sizes are presented in Figs. 6 and 7). It happened mainly due to the increasing waiting time for transactions in the ordering services.

A potential optimization solution to overcome this drawback is to process transactions in parallel with sharding. However, the effect of transaction conflicts needs to be considered. Besides, in order to achieve lower transaction latency, it is recommended to use a

lower block size (along with a lower arrival transaction rate) than the default block size. Hence, with a higher transaction arrival rate than, a higher (than the default value) block size is recommended.

Large scale IoT environment evaluations

The data collected from massive IoT devices were managed by Edge computing and IoT gateways as a middleware between the IoT sensors and Big Data systems as well as application services. The data provenance tracking was maintained to ensure the quality of shared data. The process consisted of the identity, validity, and lineage of data. Hence, edge devices could save energy for small sensors by offloading work, improving the bandwidth, and decreasing the latency. The edge devices performed preprocessing tasks and compression, resulting in significant energy saving for IoT devices. The traffic evaluations demonstrated a constant range between 20 KB/s and 30 KB/s in the idle state when there were no transactions between peer nodes and 100 KB/s during maximum load where the maximum amount of transactions were exchanged. The increase in the number of the orderer and endorser peers could improve the performance through the gossip protocol configuration. However, the results indicate that the proposed provenance model was promising for application in large-scale IoT networks with many trusted IoT sensors and devices. The generated ChainCode queries were able to retrieve 10 linked IoT records in 104 ms.

To further explore the system's performance in a large-scale IoT environment, a set of experiments were conducted with varying numbers of IoT devices connected to each edge device. By implementing a large-scale IoT environment, the impact of CPU utilization and throughput on the system were explored. The experimental environments included from 100 to 2000 IoT devices distributed equally between IoT gateways (RPi). All devices needed to be authorized before initiating communication with the network and other participants. The procedure of mutual authentication is described in "Edge computing" section. The increase in the number of IoT devices caused growth in the processing time. That was addressed by modifying the HLF configurations and adding more orderers and endorsers based on specific applications.

The system throughput and CPU utilization are illustrated in Figs. 8 and 9. Figure 8 shows a linear growth in the system throughput until it reaches the maximum load (around 1000 IoT devices). It can be seen as a result of gradually increased resource allocation by the system until all resources were fully utilized. As depicted in Fig. 9, the CPU utilization was increased, reflecting higher resource utilization by the system. After the peak point, the throughput stabilized. The CPU was mainly used during the validation phase of a generated block. Therefore, modifying the configuration of HLF in terms of the batch timeout, maximum message count, and block size can result in better performance.

Data provenance and tracking resource consumption

To further evaluate the performance of the proposed system, the federated machine learning technique [49] was considered for application across the distributed set of network participants while providing a collection of models, training, and test data sets. The framework was implemented in a way that facilitated data provenance and metadata

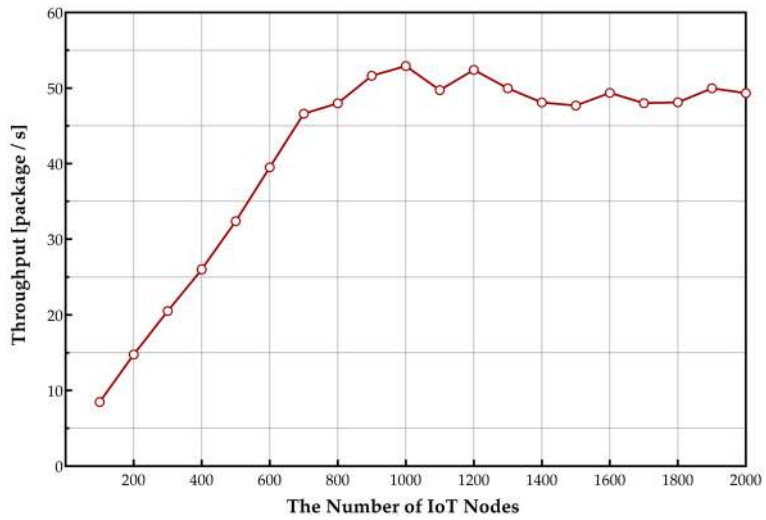

Fig. 8 Large scale IoT system throughput of edge gateways

tracking. The *ImageAI* [50] library was implemented. Training and data sets were provided in the framework. The process was initiated with storing the model. It included the following steps: data checksum computation, storing the metadata in a big data system, and maintaining the transactions by the client application library through the HLF blockchain to record the data checksum and files locations. Storing 100 models of 100 MB (the models were created using the ImageAI library) was successfully performed in around 2.3 s.

The resource consumption measurements results are presented in Figs. 10 and 11. The results show that the CPU consumption was slightly (2–3%) affected in the peer process during the model storage. The client application consumed more CPU capacity—approximately 10 to 20%. The reason for that can be found in the range of operations that needed to be handled by the client applications: computing data checksums,

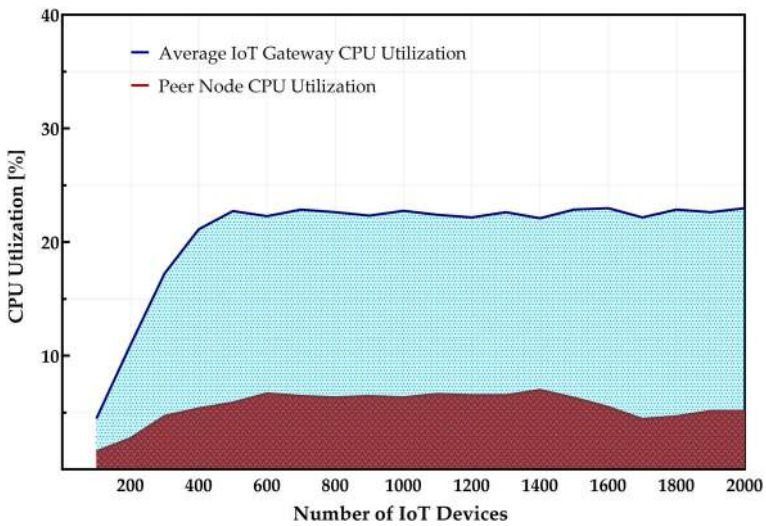

Fig. 9 IoT gateway and peer node CPU utilization

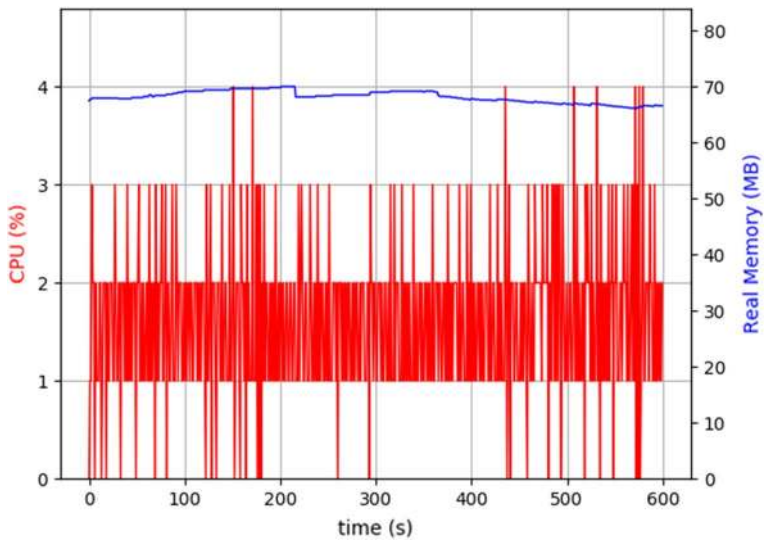

Fig. 10 Peer process CPU and memory utilization

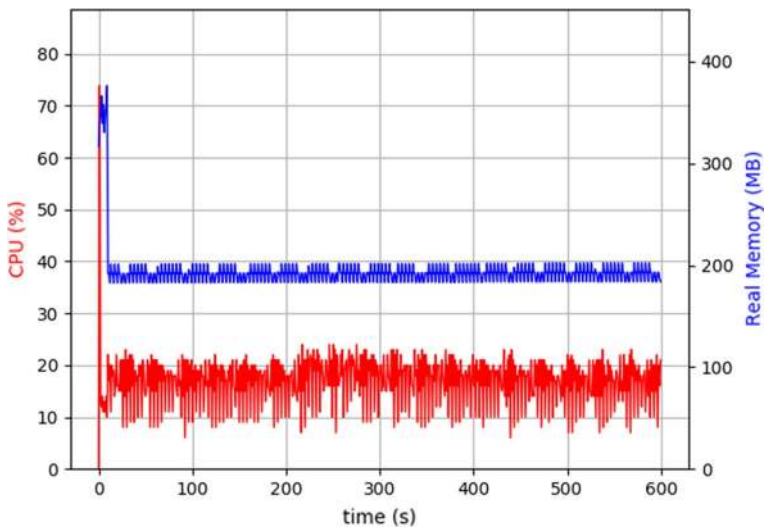

Fig. 11 Client application CPU and memory usage

storing the metadata in the Big Data system as well as storing information within the HLF blockchain. It indicates that this network model can easily be deployed on low-cost IoT devices (e.g., RPi) for real-world applications.

The client application profiling can reveal more details of the CPU consumption. Figure 12 indicates that the majority of CPU consumption by the client process was due to checksum computation and data storing in the big data system. Storing the HLF blockchain information by the client process and garbage collection occupied a small amount (some 6% and 7%, correspondingly). As mentioned in "System model and architecture" section, the client application needed to follow the sequential operations resulting in more CPU consumption. These limitations can be addressed by implementing the calculation of checksum in parallelized manner.

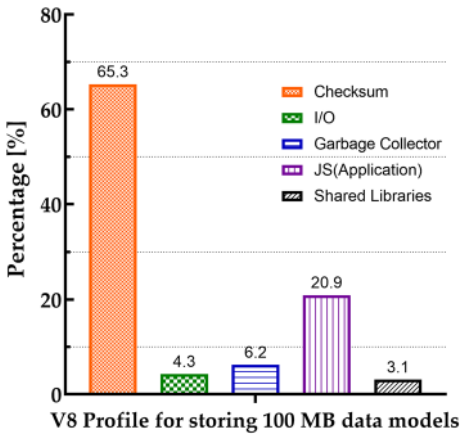

Fig. 12 Client application profiling

The network traffic measurements were conducted to explore the network overhead impact by storing the models generated by ImageAI every 2.3 s (this was for storing 100 models of 100 MB). The proposed provenance data framework was able to store the data provenance information, including data checksum, the data location, operation tags, data owner information, and some other optional parameters about the batch size. The system also provided the history of data models, tracked training datasets, and tested dataset provenance. Therefore, the lineage of successful transactions could be traced, and the model could be verified through the system. The previous measurements (associated with the utilisation of CPU, memory, and profiling) indicate that the system limitations were mainly due to the size of the file, checksum calculation, and network transfer. Figure 13 shows a low overhead for storing 100 MB data objects. Storing large files could be considered as a limitation that impacts the

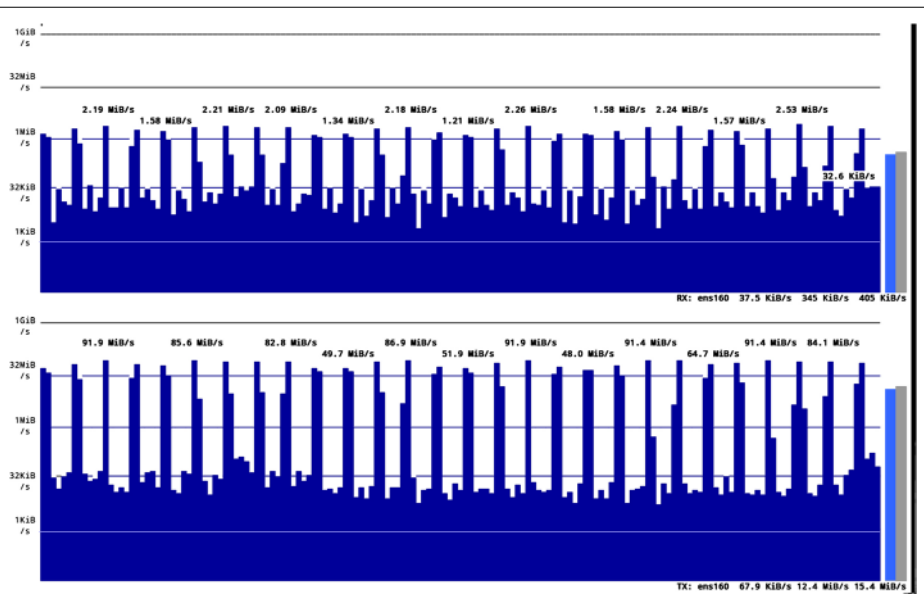

Fig. 13 Network utilization for peer process with 100 models of 100 MBs

network traffics. Hence, in such cases, the optimized solution would be to store the data provenance in the HLF. Files of large size (e.g., megabytes range) posed additional loads on the client nodes due to various resource-consuming operations such as checksum computation. The statistical results indicate no abrupt or anomalies in the network performance with changes in the system configurations. The network performance was mainly dependent on the traffic input and output. There was a progressive response to traffic changes—the observed increases in the network performance were primarily related to the growth in the network traffic thus indicating the normal network behaviour.

Conclusion

This paper proposes a blockchain-enabled secure framework for large-scale IoT data storage in a Big Data system environment. Edge computing is considered to be merged to facilitate the management of the authentications of the small IoT devices and perform data storage. A lightweight mutual authentication scheme is deployed to perform authorization and authentication of IoT devices in blockchain-based IoT applications.

The paper presents the detailed implementation of the proposed security scheme to provide the data provenance, data integrity, traceability, and auditability of IoT data in the Hadoop system as off-chain storage. The proposed model offers tamper-proof and transparent records spread across a collection of distributed peers by developing a provenance scheme using blockchain. The model also overcomes the high communication and computation overheads associated with storing large volumes of IoT data in centralized cloud storage. The proposed model eliminates the need for third-party auditing and a centralized server.

The results of the experimental research show the throughput of about 600 transactions per minute and 500 ms of the average response time. Peer and client processes were the primary resource consumers in HLF. The measurements showed about 2–3% of the CPU capacity consumption at the peer process, and approximately 10–20% at the client node. The minimum latency remained below 1 s during the experiments. However, there was an increase in the maximum latency when the sending rate reached around 200 TPS.

This study shows that the proposed scheme is a promising solution for a large-scale IoT network. Moreover, extensive experimental results demonstrate that the proposed model can be deployed to track provenance metadata with competitive throughput and latency while maintaining low computation and communication overheads. Integrating the proposed scheme with a distributed database such as Apache Cassandra to store transaction data with more detailed performance evaluations and developing a sharding-based consensus that handles the network partitions are future research directions.

The future works may include developing a framework to support more features including MQTT-based communication between blockchain, IoT sensors and Hadoop off-chain storage to store transaction data. Besides, the future works could categorize IoT data types and match them with feasible frameworks within the Hadoop ecosystem through integration with the proposed blockchain model.

Acknowledgements
Not applicable.

Authors' contributions
HHP was the main contributor of this work. He has done the literature review, experiments, data collection, prepare results, and drafted the manuscript. MR worked closely with HHP as the principal research supervisor, review, analyze, and manuscript preparation. FA and SD helped to improve the final paper. All authors read and approved the final manuscript.

Funding
This work was not funded.

Availability of data and materials
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Mechanical & Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand. [2]School of Science and Technology, Sunway University, 47500 Sunway, Selangor, Malaysia.

Received: 5 May 2021 Accepted: 20 August 2021
Published online: 30 August 2021

References

1. Marketsandmarkets: "Big Data Market by Component, Deployment Mode, Organization Size, Business Function (Operations, Finance, and Marketing and Sales), Industry Vertical (BFSI, Manufacturing, and Healthcare and Life Sciences), and Region - Global Forecast to 2025" (Accessed on 20 January 2021). online: https://www.researchandmarkets.com/r/8ww41e
2. Dedeoglu V, Jurdak R, Dorri A, Lunardi R, Michelin R, Zorzo A, Kanhere S. Blockchain technologies for iot. In: Advanced Applications of Blockchain Technology, pp. 55–89. Springer, ??? 2020.
3. Gantz J, Reinsel D. Extracting value from chaos. IDC iview. 2011;1142(2011):1–12.
4. Pouyanfar S, Yang Y, Chen S-C, Shyu M-L, Iyengar S. Multimedia big data analytics: a survey. ACM Comput Surv (CSUR). 2018;51(1):1–34.
5. Jain P, Gyanchandani M, Khare N. Enhanced secured map reduce layer for big data privacy and security. J Big Data. 2019;6(1):1–17.
6. Abouelmehdi K, Beni-Hessane A, Khaloufi H. Big healthcare data: preserving security and privacy. J Big Data. 2018;5(1):1–18.
7. Surjandari I, Yusuf H, Laoh E, Maulida R. Designing a permissioned blockchain network for the halal industry using hyperledger fabric with multiple channels and the raft consensus mechanism. J Big Data. 2021;8(1):1–16.
8. Baig MI, Shuib L, Yadegaridehkordi E. Big data adoption: state of the art and research challenges. Inf Process Manag. 2019;56(6):102095.
9. Honar Pajooh H, Rashid M, Alam F, Demidenko S. Multi-layer blockchain-based security architecture for internet of things. Sensors. 2021;21(3):772.
10. Honar Pajooh H, Rashid M, Alam F, Demidenko S. Hyperledger fabric blockchain for securing the edge internet of things. Sensors. 2021;21(2):359.
11. Deepa N, Pham Q-V, Nguyen DC, Bhattacharya S, Prabadevi B, Gadekallu TR, Maddikunta PKR, Fang F, Pathirana PN. A survey on blockchain for big data: Approaches, opportunities, and future directions. arXiv preprint arXiv:2009.00858 2020.
12. Rawat DB, Doku R, Garuba M. Cybersecurity in big data era: from securing big data to data-driven security. IEEE Trans Ser Comput. 2019.
13. Liu CH, Lin Q, Wen S. Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning. IEEE Trans Ind Inf. 2018;15(6):3516–26.
14. Xu X, Zhang X, Gao H, Xue Y, Qi L, Dou W. Become: blockchain-enabled computation offloading for iot in mobile edge computing. IEEE Trans Ind Inf. 2019;16(6):4187–95.
15. Liu G, Dong H, Yan Z, Zhou X, Shimizu S. B4sdc: a blockchain system for security data collection in manets. IEEE Trans Big Data. 2020.

16. Yang R, Yu FR, Si P, Yang Z, Zhang Y. Integrated blockchain and edge computing systems: a survey, some research issues and challenges. IEEE Commun Surv Tutor. 2019;21(2):1508–32.
17. Pahl C, El Ioini N, Helmer S, Lee B. An architecture pattern for trusted orchestration in iot edge clouds. In: 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), 2018; 63–70. IEEE.
18. Agiwal M, Roy A, Saxena N. Next generation 5g wireless networks: a comprehensive survey. IEEE Commun Surv Tutor. 2016;18(3):1617–55.
19. Wood G, et al. Ethereum: a secure decentralised generalised transaction ledger. Ethereum project yellow paper. 2014;151(2014):1–32.
20. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, Enyeart D, Ferris C, Laventman G, Manevich Y, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, 2018; 1–15.
21. Schwartz D, Youngs N, Britto A, et al. The ripple protocol consensus algorithm. Ripple Labs Inc White Paper. 2014;5(8):151.
22. Jindal A, Kumar N, Singh M. A unified framework for big data acquisition, storage, and analytics for demand response management in smart cities. Future Gener Comput Syst. 2020;108:921–34.
23. Cutting MCD. "Apache Hadoop.". http://hadoop.apache.org. Accessed 15 Feb 2021.
24. Borthakur D. The hadoop distributed file system: architecture and design. Hadoop Proj Website. 2007;11(2007):21.
25. MongoDB: "MongoDB A complete data framework.". https://www.mongodb.com/. Accessed 20 Feb 2021.
26. Spark: "Apache Spark™ is a unified analytics engine for large-scale data processing.". https://spark.apache.org/. Accessed 15 Feb 2021.
27. Storm: "Apache Storm.". https://storm.apache.org/ Accessed 15 Feb 2021.
28. Caro MP, Ali MS, Vecchio M, Giaffreda R. Blockchain-based traceability in agri-food supply chain management: a practical implementation. In: 2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany), 2018; 1–4. IEEE.
29. Javaid U, Aman MN, Sikdar B. Blockpro: Blockchain based data provenance and integrity for secure iot environments. In: Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems, 2018; 13–18
30. Liang X, Shetty S, Tosh D, Kamhoua C, Kwiat K, Njilla L. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2017; 468–477. IEEE.
31. Sigwart M, Borkowski M, Peise M, Schulte S, Tai S. A secure and extensible blockchain-based data provenance framework for the internet of things. Personal and Ubiquitous Computing, 2020;1–15.
32. Soldatos J, Kefalakis N, Hauswirth M, Serrano M, Calbimonte J-P, Riahi M, Aberer K, Jayaraman PP, Zaslavsky A, Žarko IP, et al. Openiot: Open source internet-of-things in the cloud. In: Interoperability and Open-source Solutions for the Internet of Things, 2015;13–25. Springer.
33. Yang C, Chen X, Xiang Y. Blockchain-based publicly verifiable data deletion scheme for cloud storage. J Netw Comput Appl. 2018;103:185–93.
34. Li J, Wu J, Chen L. Block-secure: blockchain based scheme for secure p2p cloud storage. Inf Sci. 2018;465:219–31.
35. Zhu L, Wu Y, Gai K, Choo K-KR. Controllable and trustworthy blockchain-based cloud data management. Future Gener Comput Syst. 2019;91:527–35.
36. Liang X, Shetty SS, Tosh D, Njilla L, Kamhoua CA, Kwiat K. Provchain: blockchain-based cloud data provenance. Blockchain for Distrib Syst Secur. 2019;69.
37. Tosh D, Shetty S, Liang X, Kamhoua C, Njilla LL. Data provenance in the cloud: a blockchain-based approach. IEEE Consumer Electr Mag. 2019;8(4):38–44.
38. Gai K, Wu Y, Zhu L, Xu L, Zhang Y. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet Things J. 2019;6(5):7992–8004.
39. Tuli S, Mahmud R, Tuli S, Buyya R. Fogbus: a blockchain-based lightweight framework for edge and fog computing. J Syst Softw. 2019;154:22–36.
40. Ren Y, Leng Y, Cheng Y, Wang J. Secure data storage based on blockchain and coding in edge computing. Math Biosci Eng. 2019;16(4):1874–92.
41. Muthanna A, A Ateya A, Khakimov A, Gudkova I, Abuarqoub A, Samouylov K, Koucheryavy A. Secure and reliable iot networks using fog computing with software-defined networking and blockchain. J Sensor Actuator Netw. 2019;8(1):15.
42. Yue D, Li R, Zhang Y, Tian W, Peng C. Blockchain based data integrity verification in p2p cloud storage. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), 2018; 561–568. IEEE
43. Wang J, Peng F, Tian H, Chen W, Lu J. Public auditing of log integrity for cloud storage systems via blockchain. In: International Conference on Security and Privacy in New Computing Environments, 2019; 378–387. Springer
44. Zhang Y, Xu C, Lin X, Shen XS. Blockchain-based public integrity verification for cloud storage against procrastinating auditors. IEEE Trans Cloud Comput. 2019.
45. Docker I. Docker. Inea].[Junio de 2017]. Disponible en: https://www.docker.com/what-docker 2017.
46. Hyperledger.: "Hyperledger fabric client sdk for node.js.". https://github.com/hyperledger/fabric-sdk-node Accessed 25 Feb 2021.
47. Locust.: Locust: an open source load testing tool. https://locust.io/. Accessed 1 Mar 2021.
48. Performance H, Group SW. "Hyperledger Blockchain Performance Metrics.". https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDF_V1.01.pdf. Accessed: 15 February 2020.
49. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol (TIST). 2019;10(2):1–19.
50. Moses O. Adams Manual: Tire Models, using the Fiala handling force model. https://github.com/OlafenwaMoses/ImageAI Accessed 1 Feb 2021.

Publisher's Note