 doi:10.20944/preprints202211.0034.v1

Article

SAUSA: Securing Access, Usage, and Storage of 3D Point Clouds Data by a Blockchain-based Authentication Network

Ronghua Xu[1], Yu Chen[1]*, Genshe Chen[2], Erik Blasch[3]

[1] Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA; rxu22@binghamton.edu (R.X.); ychen@binghamton.edu (Y.C.)
[2] Intelligent Fusion Tech, Inc., Germantown, MD 20876, USA; gchen@intfusiontech.com (G.C.)
[3] The U.S. Air Force Research Laboratory, Arlington, VA 22203, USA; erik.blasch.1@us.af.mil (E.B.)
* Corresponding author: ychen@binghamton.edu

Abstract: The rapid development of three-dimensional (3D) acquisition technology based on 3D sensors provides a large volume of data, which is often represented in the form of point clouds. Point cloud representation can preserve the original geometric information along with associated attributes in a 3D space. Therefore, it has been widely adopted in many scene-understanding-related applications such as virtual reality (VR) and autonomous driving. However, the massive amount of point cloud data aggregated from distributed 3D sensors also poses challenges for secure data collection, management, storage, and sharing. Thanks to the characteristics of decentralization and security nature, Blockchain has a great potential to improve point cloud services and enhance security and privacy preservation. Inspired by the rationales behind Software Defined Network (SDN) technology, this paper envisions SAUSA, a blockchain-based authentication network that is capable of recording, tracking, and auditing the access, usage, and storage of 3D point cloud data sets in their life-cycle in a decentralized manner. SAUSA adopts an SDN-enabled point cloud service architecture which allows for efficient data processing and delivery to satisfy diverse Quality-of-Service (QoS) requirements. A blockchain-based authentication framework is proposed to ensure security and privacy preservation in point cloud data acquisition, storage, and analytics. Leveraging smart contracts for digitizing access control policies and point cloud data on the blockchain, data owners have full control of their 3D sensors and point clouds. In addition, anyone can verify the authenticity and integrity of point clouds in use without relying on a third party. Moreover, SAUSA integrates a decentralized storage platform to store encrypted point clouds while recording references of raw data on the distributed ledger. Such a hybrid on-chain and off-chain storage strategy not only improves robustness and availability but also ensures privacy preservation for sensitive information in point cloud applications. A proof-of-concept prototype is implemented and tested on a physical network. The experimental evaluation validates the feasibility and effectiveness of the proposed SAUSA solution.

Keywords: Blockchain; Smart Contract; Point Cloud; Security; Privacy Preservation; Software-Defined Network (SND); Big Data; Assurance; Resilience

1. Introduction

With the rapid development of three-dimensional (3D) acquisition technologies, 3D sensors are increasingly available and affordable, such as LIDAR (LIght Detection And Ranging) sensors, stereo cameras, and 3D scanners. Complemented with two dimensional (2D) images, 3D data acquired by sensors demonstrates rich geometric, shape, and scale information such that it provides an opportunity for a better understanding of surrounding environments for machines [1]. In general, 3D data can be represented with different formats, such as depth image, point clouds, meshes, and volumetric grids. When compared to other 3D data formats, 3D point cloud representation preserves the original geometric information along with associate attributes in a 3D space without any discretization [1].

 doi:10.20944/preprints202211.0034.v1

Therefore, point clouds has been widely adopted in numerous application fields, including 3D scanning and modeling, environmental monitoring, agricultural and forestry, biomedical imagery, and so on [2].

Recently, deep learning (DL) on point clouds has been thriving in many scene understanding related applications, such as Virtual/Augmented Reality (VR/AR), autonomous driving, and robotics. Nevertheless, the massive amount of point cloud data aggregated from distributed 3D sensors also poses challenges for securing data collection, management, storage, and sharing. By using signal processing or neural network techniques, several efficient point cloud compression (PCC) methods [3] have been proposed to reduce the bandwidth of wireless networks or storage space of 3D point cloud raw data. However, there are still a lot of efforts to be made to achieve efficient end-to-end data delivery and optimal storage management. From the architecture aspect, conventional point cloud based applications rely on centralized cloud servers for data collection and analysis. Such a centralized manner is prone to single point failures because any successfully attacks like distributed denial-of-service (DDoS) to the control (or data) server may paralyze the entire system. Other than that, a centralized server that manages 3D sensors and stores point clouds under a distributed network environment may lead to performance bottleneck (PBN), and it is vulnerable to data breach caused by curious third parties and security threats in data acquisition, storage, and sharing process.

Because of several key features, such as separation of the control and data planes, logically centralized control, global view of the network and ability to program the network, software-defined networking (SDN) can greatly facilitate big data acquisition, transmission, storage, and processing [4]. At the same time, Blockchain has been recognized as a promising solution for security and privacy in big data applications [5] with its attractive properties, including decentralization, immutability, transparency and availability. Therefore, combining SDN and Blockchain demonstrates great potentials to revolutionize centralized point cloud systems and address aforementioned issues.

In this paper, we propose a secure-by-design networking infrastructure called SAUSA, which leverages SDN and Blockchain technologies to secure access, usage, and storage of 3D point clouds data sets in their life-cycle. SAUSA adopts a hierarchical SDN enabled service network to provide efficient and resilient point cloud applications. Network intelligence based on dynamic resource coordination and SDN controllers ensures optimal resource allocation and network configuration for point cloud applications that demand various QoS requirements. To address security issues in point cloud data collection, storage, and sharing, we design a lightweight and secure data authentication framework based on the decentralized security fabric.

By leveraging a hybrid on-chain and off-chain storage strategy, data owners can store the encrypted meta data of point clouds into distributed data storage (DDS), which is more reliable than existing solutions [6,7] that use cloud data servers to store audit proofs. In addition, encrypting meta data on DDS also protects privacy of data owners. Data owners place swarm hash of meta data and access control policy on the blockchain (on-chain storage), while original point clouds are saved by private storage servers. Thanks to transparency and auditability properties in blockchain, data owners have full control over their point cloud data, and authorized users can verify shared data without relying on any trust third-party authority. Hence, the point cloud data integrity verification is more credible in a distributed network environment.

In summary, the key contributions of this paper are highlighted as follows:

(1) A comprehensive architecture of SAUSA is introduced that consists of a hierarchical SDN enabled point cloud service network and a decentralized security fabric, and key functionalities for network traffics based on point cloud applications are described;
(2) The core design of data authentication framework is illustrated in detail, especially for workflow in data access control, integrity verification, and the structure of hybrid on-chain and off-chain storage; and

 doi:10.20944/preprints202211.0034.v1

(3) A proof-of-concept prototype is implemented and tested under a physical network that simulate the case of point cloud data sharing across multiple domains. The experimental results verify the efficiency and effectiveness of our decentralized data access authorization and integrity verification procedures.

The remainder of the paper is organized as follows: Section 2 provides background knowledge of SDN and blockchain technologies and reviews existing state-of-the-art on blockchain-based solutions to secure big data systems. Section 3 introduces rationale and system architecture of SAUSA. The details of data authentication framework are explained in Section 4. Section 5 presents prototype implementation, experimental setup, and performance evaluation. Finally, Section 6 summarizes this paper with a brief discussion on current limitations and future directions.

2. Background and Related Work

This section describes the fundamentals of the point cloud concept and explains key techniques including SDN, blockchain and smart contract. Then we introduce the state-of-the-art on decentralized solutions to secure big data acquisition, storage, and analytic.

2.1. Deep Learning on 3D Point Clouds

By providing a simpler, denser and more close-to-reality representation, 3D point clouds are prevalent in representing both static and dynamic 3D objects. By definition, a 3D point cloud is a set of points $\{P_i\}_{i=1}^{n}$ embedded in the 3D space and carrying both geometry and attribute information [2]. Given a Cartesian coordination system, the geometry information refers to the point position that can be expressed as a coordinate tuple $c_i = (x_i, y_i, z_i)$. The attribute information is used to describe the visual appearance of each point, and it may have different formats according to various user cases, such as color value tuple (R, G, B) and normal vectors (n_x, n_y, n_z).

As a dominating technology in Artificial Intelligence (AI), deep learning on point clouds has become thriving with an increasingly numbers of solutions to 3D point cloud applications, such as 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation [1]. 3D shape classification extracts a global shape embedding from the whole point cloud and then feeds it into into several fully connected layers of the neural network. In 3D object detection, a 3D object detector processes the point cloud of a scenic frame and then produces a set of detected objects with 3D bounding boxes. Given the locations of detected objects in the first frame, 3D object tracking can use the rich information of point clouds to estimates their state in subsequent frames. 3D point cloud segmentation needs the understanding of both the global geometric structure and fine-grained details of each point, and it can be classified into: semantic segmentation, instance segmentation and part segmentation [1].

2.2. Overview of SDN

The emergence of the Software-Defined Network (SDN) paradigm has attracted great interests of designing intelligent, flexible and programmable networks. As defined by Open Networking Foundation (ONF), SND refers to an emerging network architecture, where network control policies are decoupled from forwarding mechanism and are directly programmable [8]. Unlike traditional networks that are vertical integrated, the control and data planes are decoupled in SDN frameworks. As a result, control logic and network intelligence are moved to an external entity called SDN controller, while network devices simply performs forwarding decisions that are flow based rather than destination based [9]. The network is programmable through software applications running on top of the SDN controllers that logically controls the underlying network infrastructure and interacts with the upper layered management panel.

With its inherent characteristics of decoupling of control and data panels and programmability on the centralised control panel, SDN brings potential benefits in conventional network architecture and operations [8]. SDN can enhance network configuration

 doi:10.20944/preprints202211.0034.v1

and management vie unification of the control panel over heterogeneous network devices. thus, the entire network can be easily configured with programmable controllers and then dynamically optimized according to global network status. In addition, a SDN controller allows for the centralization of the control logic with global knowledge of the network state, it is promising to improve network performance with optimal utilization of underlying infrastructure. Moreover, SDN offers a convenient platform for validation of techniques and encourages innovation on next generation networks.

2.3. Blockchain and Smart Contract

From the system architecture aspect, a typical blockchain system consists of three essential components: a distributed ledger, a consensus protocol, and smart contracts [10]. Essentially, distributed ledger technology (DLT) is a type of distributed database that is shared, replicated, and maintained by all participants under a P2P networking environment. Each participant maintains a local view of the distributed ledger in the context of a distributed computing environment, and a well-established consensus allows all participants to securely reach an agreement on a global view of the distributed ledger under consideration of failures (Byzantines or crash faults). Given different consensus algorithms and network models, distributed consensus protocols are categorized into Nakamoto Consensus Protocols [11] or Byzantine Fault Tolerant (BFT) Consensus protocols [12]. From a topology aspect, blockchains can be classified into three types: public (permissionless) blockchains, private (permissioned) blockchains and consortium blockchains [13].

By using cryptographic and security mechanisms, a *smart contract* (SC) combines protocols with user interfaces to formalize and secure the relationships over computer networks [14]. Essentially, SCs are programmable applications containing predefined instructions and data stored at a unique address on the blockchain. Through exposing the public functions or application binary interfaces (ABIs), a SC acts as the trust autonomous agent between parties to perform predefined business logic functions or contract agreements under specific conditions. Owing to secure execution of predefined operational logic, unique address and public exposed ABIs, using SC provides an ideal decentralized app (Dapp) backbone to support upper level applications.

2.4. Related Work

By leveraging blockchain and deep reinforcement learning (DRL) a blockchain-enabled efficient data collection and sharing framework is proposed to provide a reliable and safety environment for data collection [15]. A distributed DRL based scheme aims to achieve the maximum data collection and ratio and geographic fairness in the long term. While Ethereum blockchain provides a tamper-proof distributed ledger to ensure ensure security and reliability of data sharing. The simulation results demonstrates the proposed scheme can prevent against attacks in data collection and sharing. However, performance of adopting blockchain is not evaluated, and storage overhead by directly storing data on the distributed ledger is not discussed.

To solve distrust issues of big data sharing on collaborative edges, a blockchain-based framework is proposed to ensure efficient and reliable data sharing across resource-limited edge nodes [16]. A green consensus mechanism called Proof-of-Collaboration (PoC) allows edge devices to mine blocks given their collaboration credits rather than computation resources. In addition, this work design a novel futile transaction filer (FTF) algorithm that offload transactions from the storage to the cache layer to reduce response time and storage overhead occupied by blockchain. Moreover, the smart contract based express transaction (E-TX) can support asynchronous validation, and hollow blocks can significantly reduce redundancy in block propagation. However, transactions encapsulating raw data are still directly stored on the distributed ledger and it brings privacy concerns.

With the popularity of edge-fog-cloud computing paradigm, verifying the integrity of data in use has become a challenge problem. Inspired by the smart contract and blockchain technology, a real-time index authentication for event-oriented surveillance video query

 doi:10.20944/preprints202211.0034.v1

system is proposed to provide a decentralized video streams security mechanism in the distributed network environment [6]. The hash value of video recordings is stored in the blockchain as immutable evidence, which is used for the authenticity of raw data in verification process. The experimental results show that the entire index authentication process incurs marginal computation overhead on service providers.

To solve issues in traditional data integrity of cloud servers, a blockchain based data integrity verification in P2P cloud storage is proposed, which allows for more open, transparent, and auditable verification of big data [7]. The raw data is divided into several shards that are stored on the private storage, while digits of shards construct hash Merkle trees that are saved on P2P cloud storage servers for data integrity verification. As root of a Merkle tree is recorded on the blockchain before uploading the data, users can verify integrity of data without relying on any third-party authority.

Combining homomorphic verification tags (HVTs) and data auditing blockchain (DAB), a decentralized big data auditing scheme is proposed for smart city environments [17]. Unlike literature [6] and [7]. data owners unloads their files and HVTs to cloud service providers (CSPs), while all auditing proofs generated by CSPs are stored into blocks of the DAB. As all historical auditing proofs cannot be tampered with, data owners or users can verify data integrity without relying on third party auditors (TPAs). The comparison shows the lower communication and computation overheads incurred in auditing process. However, storage overhead of recording auditing proofs on the DAB is not discussed.

As a decentralized storage platform that aims to address the issue of file redundancy, interplanetary file system (IPFS) is has been used to solve problems of centralize big data storage. A blockchain-based secure storage and access scheme is proposed to provide security and efficiency of electronic medical records sharing [18]. Attribute-based encryption (ABE) is used to encrypt medical data, and then encrypted data are stored in IPFS. ABE allows that only authorized users can decrypt medical data in IPFS. The hash value (data address of IPFS) of medical data are recorded in blockchain for data retrieval process and verification. Similar to scheme [18], EduRSS [19] combines blockchain, storage servers and encryption techniques to manage educational records in a decentralized manner. The encrypted original educational records are saved in distributed off-chain storage servers, while the hash information of the records is stored on the blockchain. EduRSS utilizes smart contracts to regulate the data storage and sharing process.

3. Design Rationale and System Architecture

Aiming at a self-adaptive and secure-by-design service architecture for assurance and resilience oriented 3D point cloud applications, SAUSA leverages SDN to achieve efficient resource coordination and network configuration in point cloud data processing and delivery. By combining Blockchain and distributed data storage (DDS) to build a decentralized authentication network, SAUSA is promising to guarantee security and privacy of data access, usage and storage in 3D point cloud applications.

Figure 1 demonstrates the SAUSA architecture that consists of two sub-frameworks: i) a hierarchical SDN enabled point cloud service network; ii) a decentralized security fabric based on blockchain and DDS.

3.1. Hierarchical SDN enabled Point Cloud Service Network

The left part of Figure 1 shows the hierarchy of a point cloud service network according to point cloud application stage: acquisition, aggregation and analytic. The point cloud data layer acts as an infrastructure layer including multiple domain networks, which are responsible for raw data collection, processing and delivery. In each domain, point cloud centers interconnect each others vie forwarding switches. 3D sensors generate cloud points and send them back to point cloud centers, which are actually local servers to process and store data. Given decisions made by SND controllers, forwarding switches can forward data traffic flows efficiently to satisfy QoS requirements.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

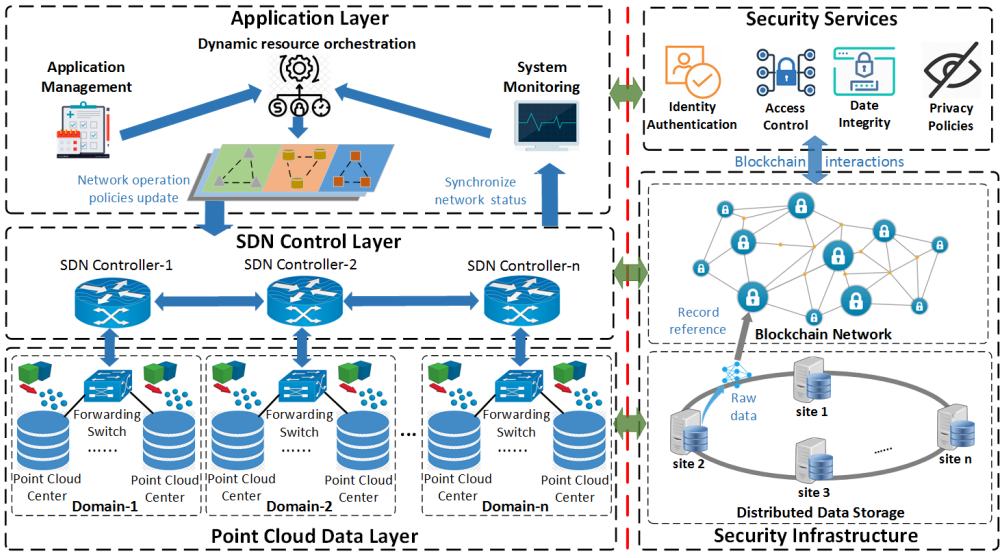

Figure 1. System Architecture of SAUSA.

Network intelligence and control logic of each domain network are performed by its SDN controller, which can be deployed on fog or cloud computing platforms. By using pre-defined southbound API, each SDN controller can either update configuration of forwarding switches to change network operations or synchronize status to have the global view of a domain network. Northbound interfaces allow a SDN controller to interact with upper-level application layer, such as providing domain network status to system monitoring and accepting network operation policies update. Therefore, these SDN controllers construct a control layer which acts as a broker between point cloud applications and fragmented domain networks, and they can provide network connectivity and data services among heterogeneous domain networks.

The application layer can be seen as "system brain" to manage physical resources of point cloud data layer with the help of SDN controllers. Application management maintains registered users and their service requirements, while system monitoring can provide the global status of the point cloud ecosystem. Given inputs from application management and system monitoring, the dynamic resource coordination adopts machine learning (ML) algorithms, which achieves fast resources (e.g, computation, network and storage) deployment and efficient service re-adjustments with QoS guarantees,

3.2. Decentralized Security Fabric

As right part of Figure 1 shows, a decentralized security fabric consists of two subsystems: i) a security services layer based on the microservice oriented architecture (MoA); ii) a fundamental security networking infrastructure atop of Blockchain and DDS. To address heterogeneity and efficiency challenges as developing and deploying security services in the distributed network environment, our security services layer adopts container technology to implement microservices for PC applications [20]. The key operations and security schemes are decoupled into multiple containerize microservices. As container is loss-coupled from remaining system with the OS-level isolation, these microservices can be independently updated, executed and terminated. Each microservice unit (or container) exposes a set of RESTful web-service APIs to users of PC applications and utilizes local ABIs to interact with SCs deployed on blockchain.

Blockchain network acts as a decentralized and trust-free platform for security services, and it uses a scalable PoW consensus protocol to ensure immutability and integrity of the on-chain data on the distributed ledger if majority (51%) miners are honest. Security mechanisms are implemented into self-executing SCs, which are deployed on the blockchain by trust oracles like system administrators. Thus, security service layer can provide secure

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

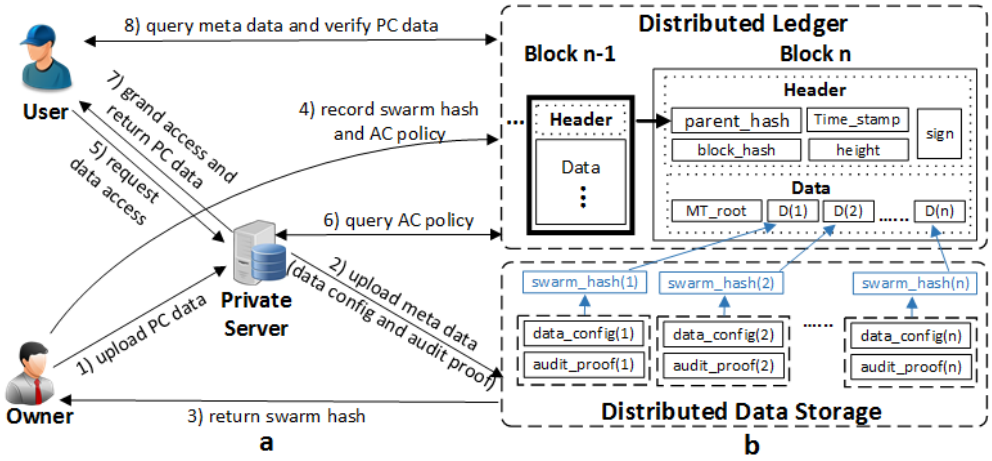

Figure 2. Illustration of blockchain-based data authentication framework. a) shows the workflow of 3D point cloud data storage, access authorization and verification. b) shows structure of the hybrid on-chain & off-chain storage

and autonomous microservices in a decentralized manner. To reduce overheads of directly recording large data on the distributed ledger, we bring DDS into security infrastructure as off-chain storage, which is built on a Swarm [21] network. The meta data of point clouds and operation logs that require heterogeneous format and various sizes are encrypted and then saved into the DDS. Raw data on DDS can be easily addressed by their references (swarm hash), which are recorded on the blockchain for audition and verification. A swarm hash has much smaller size (32 or 64 bytes) than its raw data, therefore, it is promising to improve efficiency in transaction propagation and privacy preservation without directly exposing raw data on transparent blockchain.

4. Blockchain-based Lightweight Point Cloud Data Authentication Framework

This section presents details of the decentralized and lightweight data authentication framework. SAUSA guarantees security and privacy-preservation for point clouds collection, storage and sharing. We firstly introduce participants and workflow in the framework. Then we describe the structure of hybrid on-chain and off-chain storage. Finally, we explain data access authorization and integrity verification procedures.

4.1. Data Access Control and Integrity Verification Framework

Figure 2a shows the framework of secure data access, storage and usage based on blockchain and DDS. In this framework, owners can upload point clouds generated by 3D sensors to their private server, which acts as a service provider for users of applications. By storing access control policy and audit proof into blockchain, each owner can fully control its data and the authorised user can verify data stored on the private server. The overall workflow is divided into three stages according to 3D point cloud lifecycle.

- *Data Storage*: In step 1, owner uses the secure communication channel to send encrypted point cloud data PC_i to a private server within domain network. After received point clouds in step 2, private server stores encrypted PC_i into local storage and then records meta data (e.g. configuration and audit proof) MD_i on DDS. In step 3, a site of DDS returns swarm hash as the reference to address MD_i on DDS.
- *Data Access Control*: To share point clouds with authorized users of applications, owners interact with SCs to store swarm hash of meta data and access control (AC) policy into distributed ledger (blockchain), as step 4 shows. We use a capability-based access control (CapAC) scheme [22] to implement our data access control process. In step 5, an user firstly sends data access requests to a private server that stores PC_i. Then, private server retrieves AC policy from blockchain and checks if access rights

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

assigned to user are valid, as step 6 shows. If access authentication is successful, private server return PC_i to user, as step 7 shows. Otherwise, private server denies access request without sharing data with unauthorized user.

- *Data Verification*: To audit received PC_i from private server, user queries swarm hash from blockchain and then retrieves meta data MD_i from DDB accordingly, as step 8 shows. In data verification process, user firstly checks if properties of PC_i satisfy configuration in MD_i. Then it locally calculates audit proof $AP_i^{'}$ according to PC_i and compare it with AP_i recorded in MD_i. If audit proofs are equal, the data integrity has been guaranteed. Otherwise, the data may be inconsistent with original version or corrupted during storage or sharing.

4.2. Structure of the Hybrid On-chain and Off-chain Storage

In general, a 3D model construction needs multiple segmented point clouds and each point cloud segment PC_i may have large data size and demand privacy-preservation. Thus, it's impractical to directly store point clouds into transparent blockchain for data authentication. To ensure efficient and privacy-preserving data storage and sharing, we adopt a hybrid on-chain & off-chain storage structure in data authentication framework, as shown in figure 2b. In point cloud data collection stage, meta data of point cloud segments are saved into DDS while raw data are managed by private servers. As a meta data MD_i contains data configuration (e.g., server address and properties), which is relatively small regardless of the size of the original data. In addition, an audit proof consists of integrity authenticator of a point cloud segment and a signature singed by data owner, which are byte strings with small length. Therefore, small size of meta data can greatly reduce communication cost in verification process. Furthermore, meta data are encrypted and then saved on DDS, and only authorized users are allowed to query and decrypt meta data. It is promising to protect privacy of data owners without exposing sensitive information on blockchain and DDS.

In our Swarm-based DDS, each stored meta data has an unique swarm hash as the addressable reference to actually data storage, and any change of stored data will lead to inconsistent swarm hash. Therefore, recording swarm hash on immutable distributed ledger provide non-tamperability property for meta data on DDS. To verify data integrity of a large point cloud file, swarm hash of meta data MD_i is considered as a digest $D(i)$ that is located on a leaf of Merkle tree. Then, we use such an ordered list of digests to construct a binary Merkle tree $MT\_root = BMT(D(1), D(2), ...D(N_m))$ where N_m is the number of meta data. Modifying digests or changing sequential order will lead to different root hash value MT_root of the Merkle tree. Therefore, MT_root is also stored on the distributed ledger as the data integrity proof of the entire file. In data verification process, a data user can query digests from blockchain and then parallel valid the integrity of segment data. Then, he/she can easily reconstruct Merkle tree of digests and get $MT\_root^{'}$. Finally, data integrity of the entire point cloud file can efficiently verified by comparing $MT\_root^{'}$ with MT_root on the distributed ledger.

4.3. Decentralized Data Authentication Procedures

Blockchain-based data access authorization and integrity verification procedures are presented as pseudo-code in Algorithm 1. Given a list of meta data M, data owner traverses each meta data MD_i and upload it to DDS and then appends returned swarm hash D_i to *ordered_swarm_hash*, as lines 2-6 show. Following that, data owner feeds *ordered_swarm_hash* to function $BMT()$ which will construct a binary Merkle tree and output the root hash *mk_root* (line 7). Finally, data owner call smart contract function $set\_dataAC()$ to record *mk_root* and *ordered_swarm_hash* into the distributed ledger as the public audit proof which can be uniquely addressed by *token_id* (line 8).

In data verification procedure, data user firstly uses *token_id* as the input to call smart contract function $query\_dataAC()$, which will return the public audit proof information stored on the blockchain (line 10). Regarding token validation, data user performs func-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

Algorithm 1 The data access authorization and integrity verification procedures.

```
1:  procedure: authorize_data(token_id, M)
2:      ordered_swarm_hash = []
3:      for MD_i in M do
4:          D_i ← upload_data(MD_i)
5:          ordered_swarm_hash.append(D_i)
6:      end for
7:      mt_root ← BMT(ordered_swarm_hash)
8:      receipt ← Contract.set_dataAC(token_id, mt_root, ordered_swarm_hash)
9:  procedure: verify_data(token_id)
10:     mt_root, ordered_swarm_hash ← Contract.query_dataAC(token_id)
11:     mt_root' ← BMT(ordered_swarm_hash)
12:     if mt_root' ≠ mt_root then
13:         return False
14:     end if
15:     MD = []
16:     for D_i in ordered_swarm_hash do
17:         MD_i ← download_data(D_i)
18:         if MD_i == NULL then
19:             return False, NULL
20:         end if
21:         MD.append(MD_i)
22:     end for
23:     return True, MD
```

tion $BMT()$ on received *ordered_swarm_hash* to recover root hash mt_root', then check if mt_root' is consistent with audit proof mt_root. If validation fails, directly return false result. Otherwise, it goes ahead to meta data verification. Given received *ordered_swarm_hash*, data user traverses each digest D_i which is used to download meta data MD_i from DDS. Any wrong digest or corrupted meta data will lead to $NULL$ result returned by function $download\_data()$. Finally, a valid list of meta data is returned only if all meta data can be successfully retrieved, as lines 16-23 show.

5. Experimental Results and Evaluation

In this section, experimental configuration based on a proof-of-concept prototype implementation is described. Following that, we evaluate performance of running SAUSA based on numerical results, which especially focus on the impact of blockchain on system performance. Finally, comparative evaluation among previous work highlights the main contributions of SAUSA in terms of lightweight blockchain design, performance improvement, security and privacy properties.

5.1. Prototype Implementation

We use Python language to implement a proof-of-concept prototype including client & server applications and microservices. A micro-framework called Flask [23] is used to develop RESTful APIs for applications and microservices. We use standard python library cryptography [24] to develop all security primitives, such as digital signature, symmetric cryptography (Fernet) and hash function (SHA-256). Solidity [25] is used for smart contracts implementation and test, and all SCs are deployed on a private Ethereum test network.

The experimental infrastructure worked under a physical local area network (LAN) environment and included a cloud server and several desktops and Raspberry Pi (Rpi) boards. Figure 3 shows the experimental setup for our prototype validation. A desktop emulates the private server that stores point clouds data managed by data owner, while a Rpi simulates client (user) that request data access. A private Ethereum network consists of 6 miners that are deployed on the cloud server as 6 containers separately, and each

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

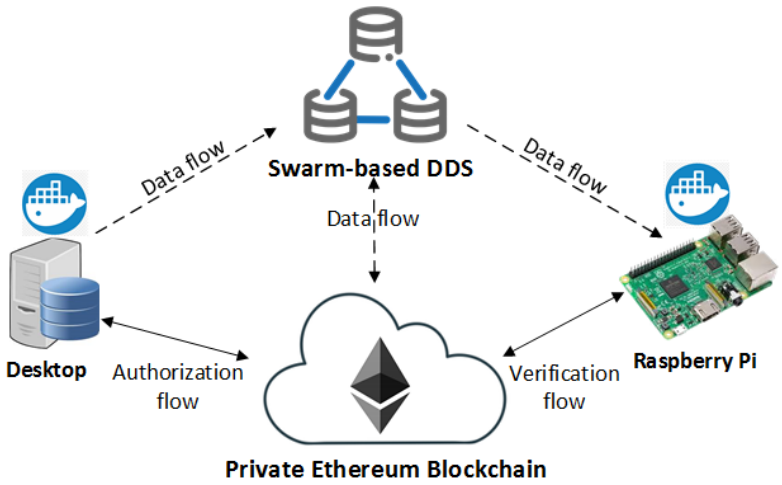

Figure 3. The experimental setup and network configuration.

Table 1. Configuration of Experimental Nodes.

Device	Cloud Server	Desktop	Raspberry Pi 4 Model B
CPU	Intel(R) Xeon(R) Gold 5220R CPU @ 2.20 GHz (96 cores)	Intel Core TM i5-3470 (4 cores), 3.2 GHz	Broadcom ARM Cortex A72 (ARMv8), 1.5 GHz
Memory	512 GB DDR4	16 GB DDR3	4 GB SDRAM
Storage	4 TB HHD	500GB HHD	64GB (microSD)
OS	Ubuntu 20.04	Ubuntu 20.04	Raspbian (Jessie)

containerized miner is assigned one cpu core. While other microservice containers that are deployed on desktop and RPis work in a light-node mode without mining blocks. All participants use Go-Ethereum [26] as client applications to interact with smart contracts on the private Ethereum network. Regarding a Swarm-based DDS, we built a private Swarm test network consisting of five desktops as service sites. Table 1 describes devices that are used to build the experimental testbed.

5.2. Performance Evaluation

This section evaluates the performance of executing operations in data authorization and verification. In a data authorization process, desktop launches a transaction that encapsulates swarm hash of meta data to the blockchain, and then states of SC can be updated until a block containing transactions is committed by miners. Thus, we evaluate end-to-end latency and gas usage during a successful data authorization operation. According to Algorithm 1, the whole data integrity verification procedure is divided into three steps: 1) client (Rpi or desktop) queries data token containing swarm hash of meta data and merkle root from the blockchain; 2) client validates merkle root and swarm hash in data token; and 3) client retrieves meta data from DDS and verifies them. Therefore, we evaluate the processing time of individual step on different platforms as changing the number of meta data (N_m). Finally, we analyze computation overheads incurred by retrieving meta data from DDS and performing symmetric encryption on meta data. We conducted 50 Monte Carlo test runs for each test scenarios and used the averages to measure results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

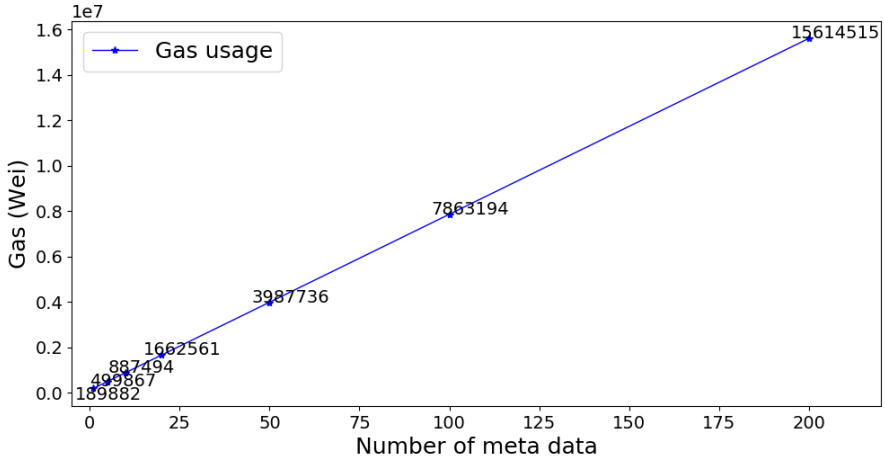

Figure 4. Gas usage in data authorization.

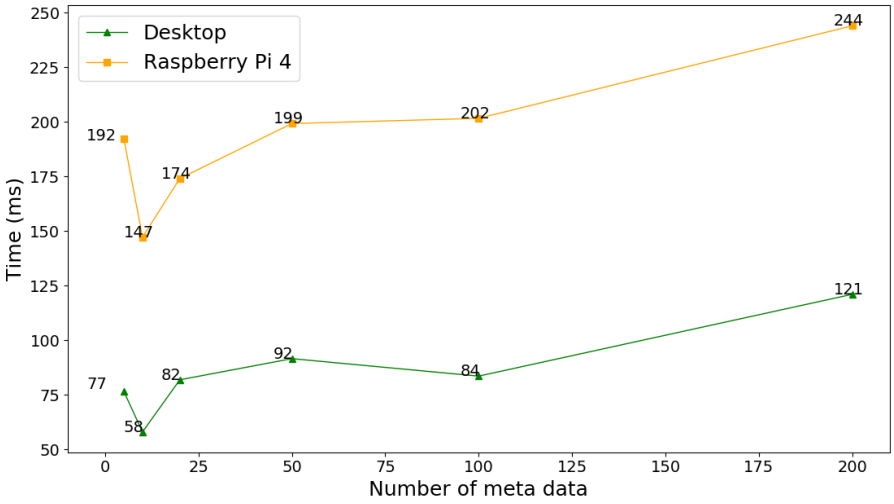

Figure 5. Latency by data token query on different platforms.

5.2.1. End-to-End Latency and Gas Usage by Data Authorization

We scale up N_m in data authorization scenarios to evaluate how the size of the ordered list of digests (swarm hash) impacts the performance. As a transaction committed time is greatly influenced by the blockchain confirmation time, we observe that all data authorization operations with different N_m demonstrate almost the similar end-to-end latency (about 4 sec) in our private Ethereum network. Regarding various computation complexity and processed data required by SC, gas used by transactions may vary. Figure 4 shows the gas usage by data authorization transactions as N_m increases. The longer ordered list of digests, the more gas used by per transaction that store data on the blockchain. Hence, recording swarm hash rather than meta data or even raw data on the distributed ledger can greatly reduce gas consumption of blockchain transaction.

5.2.2. Processing time by Data Verification

Figure 5 shows average delays that evaluate how long a data token query function of SC can be successfully handled by the client as N_m increases from 5 to 200. Regarding a larger N_m, the query token procedure of SC needs more computation resources to process data on the distributed ledger. Thus, the delays of querying a data token on both platforms are linear scale to N_m with the same gain. Due to different computation resources, the processing time of data token query on Rpi is almost double than desktop.

Figure 6 shows computation overheads by validating token data on the client side as N_m changes. The data token data validation requires to reconstruct the structure of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

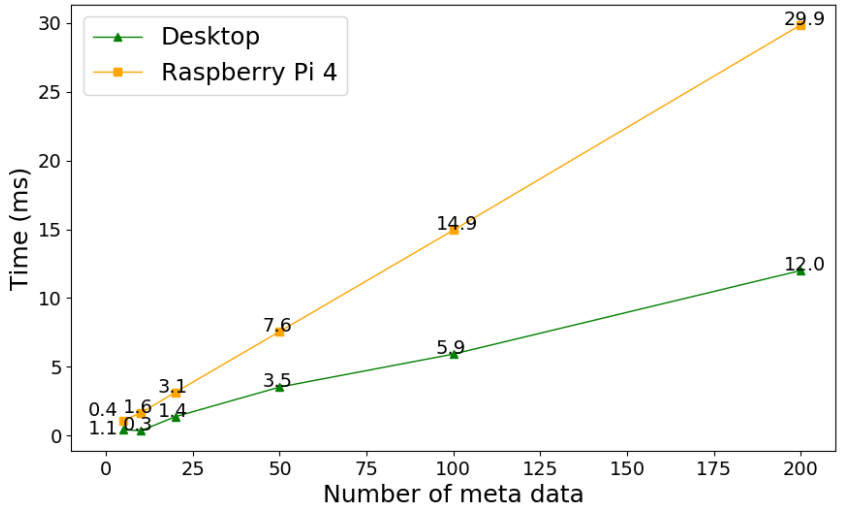

Figure 6. Processing time by data token validation on different platforms.

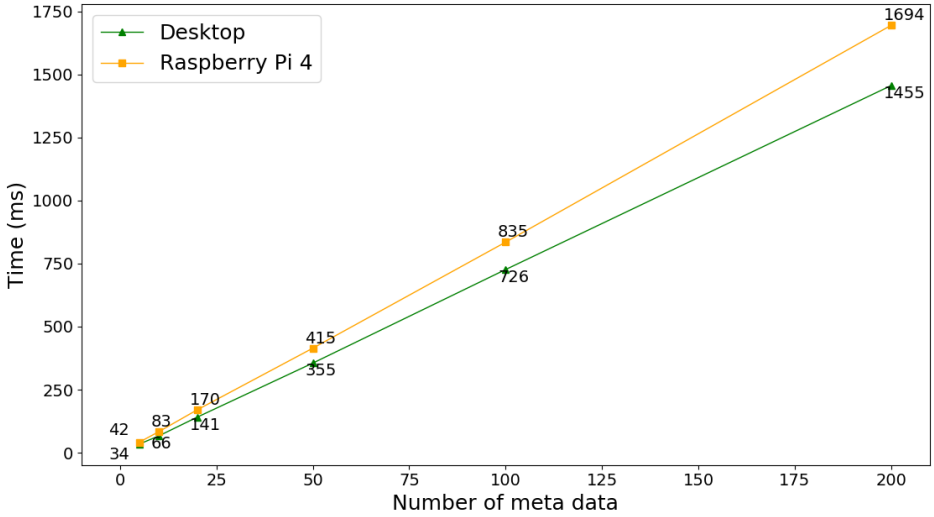

Figure 7. Processing time by meta data verification on different platforms.

binary Merkle tree of the ordered list of swarm hash, which produces traversal complexity $\mathcal{O}(N_m)$. Then root hash can be used as the fingerprint for the entire meta data to check inconsistencies, which needs $\mathcal{O}(1)$ computation complexity. Finally, computation overheads incurred by verifying token data are linear scale to N_m. Computing the root hash of binary Merkle tree demands intensive hash operations such that computation power of client machines dominates performance of data token validation. Therefore, larger N_m in data token validation brings more delays on Rpi than desktop. However, the impact is almost marginal in our test scenarios that $N_m \leq 200$.

Figure 7 shows the processing time of verifying meta data on the client side as N_m increases. In meta data verification stage, a client uses swarm hash list in data token to sequentially retrieve N_m meta data from DDS, which needs communication complexity $\mathcal{O}(N_m)$. Regarding the fixed bandwidth of test network, increasing N_m allows for larger Round Trip Time (RTT) and more computation resources in meta data transmission. As a result, delays of verifying a batch of meta data are linear scale to N_m. Unlike desktop, Rpi has limited computation resource to handle per data transmission. Therefore, Rpi takes longer time for verifying the same amount of meta data than desktop does.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 November 2022 doi:10.20944/preprints202211.0034.v1

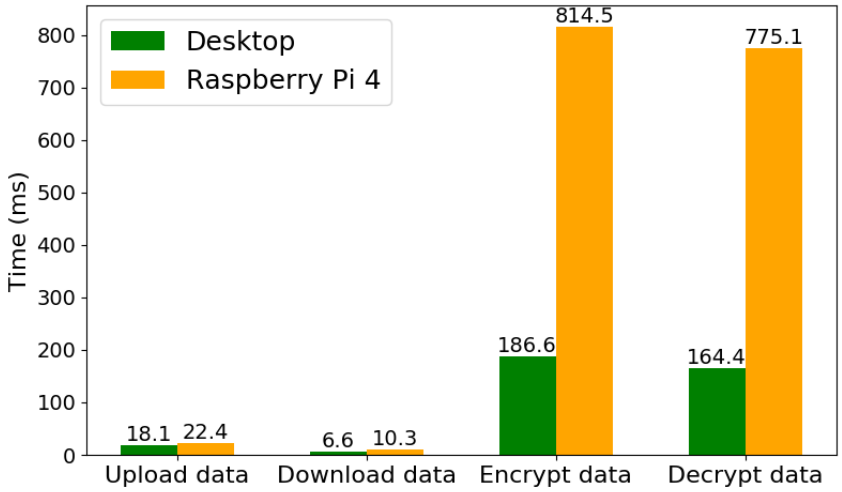

Figure 8. Processing time of meta data operations: accessing Swarm and symmetric encryption.

Table 2. Comparison among existing blockchain-based solutions.

Scheme	Blockchain	Storage	Performance	Security	Privacy
[15]	×	DLT	×	√	×
[16]	Green Blockchain	DLT	√	√	×
[6]	Ethereum	Fog Server	√	√	×
[7]	×	DDS	√	√	×
[17]	×	DLT	×	√	×
[18]	×	DDS	√	√	×
[19]	Ethereum	Storage Server	√	√	√
SAUSA	Ethereum	DDS	√	√	√

5.2.3. Computation Cost by Preserving Meta Data Privacy

In our test scenario, the average size of meta data file is about 2 KB. Figure 8 shows the process time of accessing data from (to) DDS and executing encryption over a meta data file on the client side. The delays incurred by uploading a meta data file to swarm network and then downloading it from a service site are almost the same on desktop and Rpi. However, RPi takes longer process time to encrypt and decrypt data than desktop does due to limited computation and memory resources. Compared with swarm operations, performing encryption algorithms on meta data brings extra overheads in data verification process on both platforms. As a trade-off, using encrypted meta data to ensure privacy preservation is inevitable at the cost of the longer latency in service process.

5.3. Comparative Evaluation

Table 2 presents the comparison between our SAUSA and previous blockchain-based solutions to big data applications. The symbol √ indicates that the scheme guarantees the properties, and × indicates the opposite case. Unlike existing solutions that lack details on optimal network framework for QoS or evaluations on the impact of applying blockchain into big data applications, we illustrate a comprehensive system architecture, along with details on SDN-based service network and lightweight data authentication framework. We especially evaluate performance (e.g,. network latency, processing time and computation overheads) of the blockchain enabled security mechanism in data access authentication and integrity verification process.

 doi:10.20944/preprints202211.0034.v1

Regarding storage optimization and privacy preservation for point cloud data sharing, a hybrid on-chain and off-chain data storage structure not only reduces communication and storage overheads by avoiding directly saving large volumes of raw data or audit proofs into blockchain transactions, and it also protects sensitive information by only exposing references of encrypted meta data on the transparent distributed ledger as the fingerprint proof. Unlike existing solutions that rely on a centralized off-chain storage (e.g., centralized fog server or storage server) to store audit proofs, using a decentralized Swarm network as the off-chain storage is promising to enhance robustness (availability and recoverability) for point cloud data sharing in multi-domain applications.

6. Conclusions and Future Work

This paper presents SAUSA, which combines SDN and blockchain technology to support efficiency, assurance and resilience oriented point cloud applications. The hierarchical SDN enabled service network can provide efficient resource coordination and network configuration to satisfy QoS of point cloud applications. A lightweight data authentication framework atop of blockchain and DDS aims to secure 3D point cloud data access, usage and storage in a decentralized manner. The experimental results based on a prototype implementation demonstrate the effectiveness and efficiency of our SAUSA. However, there are open questions which need to be addressed before applying the SAUSA to real-world 3D point cloud scenarios. We leave these limitations to our future works:

(1) SAUSA uses Ethereum to build blockchain network that ensures security and scalability in open access networks. However, PoW mining brings unsustainable energy consumption, longer transaction committed latency and lower throughput. Thus, it is not suitable for time sensitive applications. Lightweight blockchain designs, like Microchain [27], are promising to optimize computation utilization and improve performance in terms of end-to-end latency and transaction throughput. Our ongoing efforts includes validating SAUSA in a real-world point cloud scenario and investigation on integration of Microchain to reduce data authorization latency.
(2) This paper focuses on decentralized security scheme implementation and validation, however, there are still unanswered questions and challenges about networking service intelligence in point cloud applications. In future work, we will investigate SDN controllers and virtual network functions (VNFs) to efficiently manage network and storage resources within each domain, and evaluate the system performance and security properties according to various attack scenarios.

Funding: This research was partially funded by the United State National Science Foundation (NSF) under the grant CNS-2141468.

Acknowledgments: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. government.

Abbreviations

The following abbreviations are used in this manuscript:

 doi:10.20944/preprints202211.0034.v1

ABI	Application Binary Interfaces
AC	Access Control
AI	Artificial Intelligence
AR	Augmented Reality
BFT	Byzantine Fault Tolerant
DApp	Decentralized App
DDS	Distributed Data Storage
DDoS	Distributed Denial-of-Service
DL	Deep Learning
DLT	Distributed Ledger Technology
IoT	Internet of Things
IPFS	Interplanetary File System
LIDAR	LIght Detection And Ranging
ML	Machine Learning
MoA	Microservice Oriented Architecture
ONF	Open Networking Foundation
P2P	Peer-to-Peer
PBN	Performance Bottleneck
PC	Point Cloud
QoE	Quality-of-Experience
QoS	Quality-of-Service
RRT	Round Trip Time
SC	Smart Contract
SDN	Software-Defined Networking
SPF	Single Point of Failures
VR	Virtual Reality

References

1. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence* **2020**, *43*, 4338–4364.
2. Cao, C.; Preda, M.; Zaharia, T. 3D point cloud compression: A survey. In Proceedings of the The 24th International Conference on 3D Web Technology, 2019, pp. 1–9.
3. Bui, M.; Chang, L.C.; Liu, H.; Zhao, Q.; Chen, G. Comparative Study of 3D Point Cloud Compression Methods. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021, pp. 5859–5861.
4. Cui, L.; Yu, F.R.; Yan, Q. When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE network* **2016**, *30*, 58–65.
5. Deepa, N.; Pham, Q.V.; Nguyen, D.C.; Bhattacharya, S.; Prabadevi, B.; Gadekallu, T.R.; Maddikunta, P.K.R.; Fang, F.; Pathirana, P.N. A survey on blockchain for big data: approaches, opportunities, and future directions. *Future Generation Computer Systems* **2022**.
6. Nikouei, S.Y.; Xu, R.; Nagothu, D.; Chen, Y.; Aved, A.; Blasch, E. Real-time index authentication for event-oriented surveillance video query using blockchain. In Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2). IEEE, 2018, pp. 1–8.
7. Yue, D.; Li, R.; Zhang, Y.; Tian, W.; Peng, C. Blockchain based data integrity verification in P2P cloud storage. In Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2018, pp. 561–568.
8. Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials* **2014**, *17*, 27–51.
9. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE* **2014**, *103*, 14–76.
10. Xu, R.; Nagothu, D.; Chen, Y. Decentralized video input authentication as an edge service for smart cities. *IEEE Consumer Electronics Magazine* **2021**, *10*, 76–82.
11. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.
12. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* **1982**, *4*, 382–401.
13. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet of Things Journal* **2018**, *6*, 2188–2204.
14. Szabo, N. Formalizing and securing relationships on public networks. *First monday* **1997**.
15. Liu, C.H.; Lin, Q.; Wen, S. Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning. *IEEE Transactions on Industrial Informatics* **2018**, *15*, 3516–3526.
16. Xu, C.; Wang, K.; Li, P.; Guo, S.; Luo, J.; Ye, B.; Guo, M. Making big data open in edges: A resource-efficient blockchain-based approach. *IEEE Transactions on Parallel and Distributed Systems* **2018**, *30*, 870–882.

17. Yu, H.; Yang, Z.; Sinnott, R.O. Decentralized big data auditing for smart city environments leveraging blockchain technology. *IEEE Access* **2018**, *7*, 6288–6296.
18. Sun, J.; Yao, X.; Wang, S.; Wu, Y. Blockchain-based secure storage and access scheme for electronic medical records in IPFS. *IEEE Access* **2020**, *8*, 59389–59401.
19. Li, H.; Han, D. EduRSS: A blockchain-based educational records secure storage and sharing scheme. *IEEE Access* **2019**, *7*, 179273–179289.
20. Xu, R.; Zhai, Z.; Chen, Y.; Lum, J.K. BIT: A blockchain integrated time banking system for community exchange economy. In Proceedings of the 2020 IEEE International Smart Cities Conference (ISC2). IEEE, 2020, pp. 1–8.
21. Swarm. https://ethersphere.github.io/swarm-home/. Accessed on September 2022.
22. Xu, R.; Chen, Y.; Blasch, E.; Chen, G. Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot. *Computers* **2018**, *7*, 39.
23. Flask: A Pyhon Microframework. [Online]. Available: https://flask.palletsprojects.com/. Accessed on September 2022.
24. pyca/cryptography documentation. [Online]. Available: https://cryptography.io/. Accessed on September 2022.
25. Solidity. https://docs.soliditylang.org/en/v0.8.13/. Accessed on September 2022.
26. Go-ethereum. https://ethereum.github.io/go-ethereum/. Accessed on September 2022.
27. Xu, R.; Chen, Y.; Blasch, E. Microchain: A Light Hierarchical Consensus Protocol for IoT Systems. In *Blockchain Applications in IoT Ecosystem*; Springer, 2021; pp. 129–149.