future internet

MDPI

Article

SAUSA: Securing Access, Usage, and Storage of 3D Point CloudData by a Blockchain-Based Authentication Network

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

[1] Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA
[2] Intelligent Fusion Tech, Inc., Germantown, MD 20876, USA
[3] The U.S. Air Force Research Laboratory, Arlington, VA 22203, USA
* Correspondence: ychen@binghamton.edu

Abstract: The rapid development of three-dimensional (3D) acquisition technology based on 3D sensors provides a large volume of data, which are 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, Blockchain has great potential to improve point cloud services and enhance security and privacy preservation. Inspired by the rationales behind the 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 datasets in their life-cycle in a decentralized manner. SAUSA adopts an SDN-inspired 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 (SDN); big data; assurance; resilience

Citation: Xu, R.; Chen, Y.; Chen, G.; Blasch, E. SAUSA: Securing Access, Usage, and Storage of 3D Point CloudData by a Blockchain-Based Authentication Network. *Future Internet* **2022**, *14*, 354. https://doi.org/10.3390/fi14120354

Academic Editors: Christoph Stach and Clémentine Gritti

Received: 1 November 2022
Accepted: 25 November 2022
Published: 28 November 2022

1. Introduction

With the rapid development of three-dimensional (3D) acquisition technologies, 3D sensors are increasingly available and affordable, such as light detection and ranging (LIDAR) sensors, stereo cameras, and 3D scanners. Complemented with two-dimensional (2D) images, 3D data acquired by sensors demonstrate rich geometric, shape, and scale information such that they provide 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 images, 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].

Therefore, point clouds have been widely adopted in numerous application fields, including 3D scanning and modeling, environmental monitoring, agricultural and forestry, bio-medical 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 the storage space of 3D point cloud raw data. However, there are still many efforts 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 successful attacks such as 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 bottlenecks (PBNs), and it is vulnerable to data breaches caused by curious third parties and security threats in data acquisition, storage, and sharing process.

Because of several key features, such as the separation of the control and data planes, logically centralized control, the global view of the network, and the 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 potential to revolutionize centralized point cloud systems and address the 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 datasets 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 the privacy of data owners. Data owners place the Swarm hash of meta-data and the access control policy on the Blockchain (on-chain storage), while the original point clouds are saved by private storage servers. Thanks to the transparency and auditability properties of Blockchain, data owners have full control over their point cloud data, and authorized users can verify shared data without relying on any trusted 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) The comprehensive architecture of SAUSA is introduced, which 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 the data authentication framework is illustrated in detail, especially for the workflow in data access control, integrity verification, and the structure of hybrid on-chain and off-chain storage;

(3) A proof-of-concept prototype is implemented and tested under a physical network that simulates 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 the background knowledge of SDN and Blockchain technologies and reviews the existing state-of-the-art on Blockchain-based solutions to secure big data systems. Section 3 introduces the rationale and system architecture of SAUSA. The details of data the authentication framework are explained in Section 4. Section 5 presents the prototype implementation, experimental setup, performance evaluation, and security analysis. 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 contracts. 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, which 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 been thriving with an increasing number of solutions to 3D point cloud applications, and typical examples are 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation [1]. Regarding 3D shape classification, the whole point cloud file is used to extract a global shape embedding, which is then input into several fully connected layers of the neural network to achieve the classification task [8]. In 3D object detection scenarios, a 3D object detector firstly processes the point cloud of a frame, and then, it produces a set of detected objects with 3D bounding boxes [9]. As a 3D object detection algorithm can detect the locations of target objects in the first frame, 3D object tracking methods can use the embedded rich information of point clouds to estimate their state in subsequent frames [10]. Given the understanding of the global geometric structure and fine-grained details of each point, 3D point cloud segmentation methods can be classified into three types: 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 interest in designing intelligent, flexible. and programmable networks. As defined by the Open Networking Foundation (ONF), SDN refers to an emerging network architecture, where network control policies are decoupled from the forwarding mechanism and are directly programmable [11]. Unlike traditional networks that are vertically 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 the SDN controller, while network devices simply make forwarding decisions that are flow-based rather than destination-based [12]. The network is programmable through software applications running on top of the SDN controllers that logically control the underlying network infrastructure and interact with the upper-layer management panel.

With its inherent characteristics of decoupling the control and data panels and programmability on the centralized control panel, SDN brings potential benefits in conven-

tional network architecture and operations [11]. SDN can enhance network configuration and management by using the 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 the global network status. In addition, an SDN controller allows for the centralization of the control logic with global knowledge of the network state, and it is promising to improve network performance with optimal utilization of the underlying infrastructure. Moreover, SDN offers a convenient platform for the validation of techniques and encourages the innovation of 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. 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 the consideration of failures (Byzantines or crash faults). Given different consensus algorithms and network models, distributed consensus protocols are categorized into Nakamoto consensus protocols [13] or Byzantine fault-tolerant (BFT) consensus protocols [14]. From the topology aspect, Blockchains can be classified into three types: public (permissionless) Blockchains, private (permissioned) Blockchains, and consortium Blockchains [15].

Thanks to the cryptographic and secure computing schemes, a *smart contract* (SC) brings programmability to the Blockchain by integrating protocols with user interfaces to formalize and secure the relationships of participants over computer networks [16]. Essentially, SCs are programmable applications containing predefined instructions and data, and they are compiled and saved in the addressable storage of the Blockchain. Through exposing a set of public functions or application binary interfaces (ABIs), an SC acts as the autonomous trusted agent between parties to perform predefined business logic functions or contract agreements under specific conditions. Owing to the secure execution of the predefined functional logic, global unique address, and open-access ABIs, the SC is an ideal candidate to implement decentralized applications (Dapps) under dynamic, heterogeneous, and distributed network environments.

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 safe environment for data collection [17]. A distributed DRL-based scheme aims to achieve the maximum data collection and ratio and geographic fairness in the long term, while the Ethereum Blockchain provides a tamper-proof distributed ledger to ensure the security and reliability of data sharing. The simulation results demonstrated that the proposed scheme can prevent against attacks in data collection and sharing. However, the performance adopting Blockchain has not been evaluated, and the storage overhead by directly storing data on the distributed ledger was not discussed.

To solve the distrust issues of big data sharing on collaborative edges, a Blockchain-based framework was proposed to ensure efficient and reliable data sharing across resource-limited edge nodes [18]. A green consensus mechanism called proof-of-collaboration (PoC) allows edge devices to mine blocks given their collaboration credits, rather than their computational resources. In addition, this work designed a novel futile transaction filer (FTF) algorithm that offloads transactions from the storage to the cache layer to reduce the response time and storage overhead occupied by the Blockchain. Moreover, the smart-contract-based express transaction (E-TX) can support asynchronous validation, and hollow blocks can significantly reduce the redundancy in block propagation. However, transactions

encapsulating raw data are still directly stored on the distributed ledger, and this brings privacy concerns.

With the popularity of the edge–fog–cloud computing paradigm, verifying the integrity of data in use has become a challenging problem. Inspired by the smart contract and Blockchain technology, a real-time index authentication for an event-oriented surveillance video query system is proposed to provide a decentralized video stream 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 the verification process. The experimental results showed that the entire index authentication process incurs marginal computational overhead for service providers.

To solve the issues of the 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 are divided into several shards, which are stored in private storage, while the digits of the shards construct the hash Merkle trees, which are saved on P2P cloud storage servers for data integrity verification. As the root of a Merkle tree is recorded on the Blockchain before uploading the data, users can verify the integrity of the data without relying on any third party authority.

Combining homomorphic verification tags (HVTs) and the data-auditing Blockchain (DAB), a decentralized big data auditing scheme is proposed for smart city environments [19]. Unlike [6,7], the data owners unload their files and HVTs to cloud service providers (CSPs), while all auditing proofs generated by the CSPs are stored into the blocks of the DAB. As all historical auditing proofs cannot be tampered with, data owners or users can verify the data integrity without relying on third party auditors (TPAs). The comparison shows the lower communication and computational overheads incurred in the auditing process. However, the storage overhead of recording auditing proofs on the DAB was not discussed.

As a decentralized storage platform that aims to address the issue of file redundancy, the Interplanetary File System (IPFS) has been used to solve the problems of centralized big data storage. A Blockchain-based secure storage and access scheme is proposed to provide the security and efficiency of electronic medical record sharing [20]. Attribute-based encryption (ABE) is used to encrypt medical data, and then, the encrypted data are stored in the IPFS. ABE allows only authorized users to decrypt medical data in the IPFS. The hash values (data address of the IPFS) of the medical data are recorded in the Blockchain for data retrieval process and verification. Similar to the scheme in [20], EduRSS [21] 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.

To comply with the privacy requirement of the General Data Protection Regulation (GDPR), which allows users to rectify or even erase their own data, several solutions have been proposed to delete and update data on the Blockchain. A redactable Blockchain based on hash function modification is proposed to re-write or compress the on-chain data on the append-only distributed ledger [22]. Due to secret trapdoor information, chameleon hash functions [23] can efficiently find hash collisions, which allow for redactable blocks without breaking the hash chain. To enable redactable off-chain data over the IPFS, a delegated content erasure is proposed to enforce complete content removal across the entire network [24]. The proposed protocol relies on a "proof-of-ownership" to ensure anonymous and censorship-resistant off-chain data storage, such that only a user is allowed to delete its own contents. Unlike the above redactable solutions, a pseudonymization-based approach [25] is proposed to satisfy GDPR as integrating with the Blockchain. The pseudonymization uses cryptographic hash functions for encrypting the date or pseudonymous identities for anonymity. Therefore, only users who have encryption keys of the pseudonymization can decrypt data or even eliminate content.

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 the security and privacy of data access, usage, and storage in 3D point cloud applications.

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

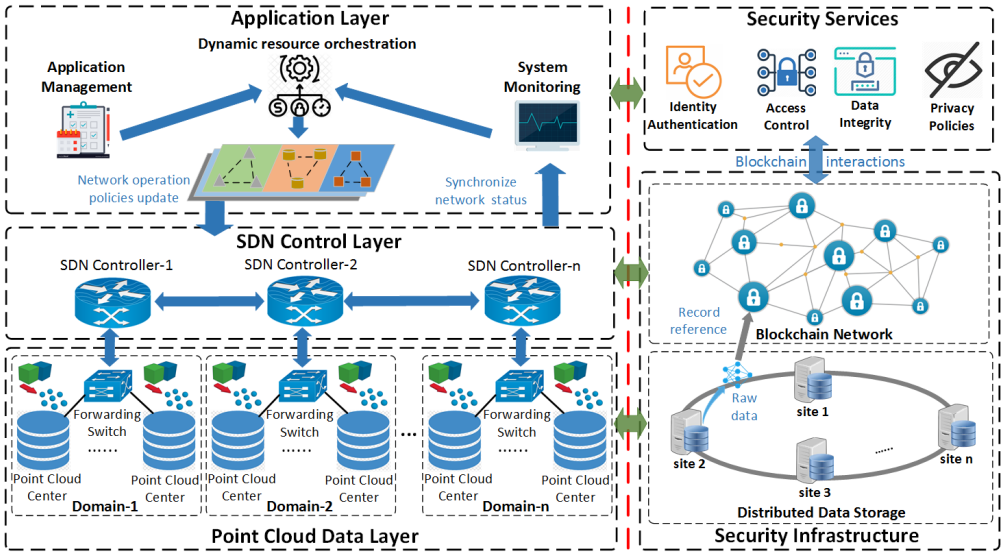

Figure 1. System architecture of SAUSA.

3.1. Hierarchical SDN-Enabled Point Cloud Service Network

As a potential technology to improve network performance and reduce management cost, the rationale of SDN is utilized to design a conceptual network architecture for multi-domain PC applications. Since this paper focuses on the Blockchain-based authentication network architecture, the key components and the workflow of the SDN are briefly described. The detailed SDN designs will be presented in our future work. 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 with each others via forwarding switches. The 3D sensors generate cloud points and send them back to the point cloud centers, which are actually local servers, to process and store the data. Given the decisions made by the SDN controllers, the forwarding switches can forward the data traffic flows efficiently to satisfy the QoS requirements.

The network intelligence and control logic of each domain network are performed by the SDN controller, which can be deployed on fog or cloud computing platforms. By using a pre-defined southbound API, each SDN controller can either update the configuration of forwarding switches to change the network operations or synchronize the status to have the global view of a domain network. Northbound interfaces allow an SDN controller to interact with the upper-level application layer, such as providing domain network status to the system monitoring and accepting the 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 a "system brain" to manage the physical resources of the point cloud data layer with the help of SDN controllers. The application management maintains registered users and their service requirements, while the system monitoring can provide the global status of the point cloud ecosystem. Given the inputs from application management and system monitoring, the dynamic resource coordination adopts machine learning (ML) algorithms, which achieve fast resource (e.g., computation, network, and storage) deployment and efficient service re-adjustments with QoS guarantees.

3.2. Decentralized Security Fabric

As the right part of Figure 1 shows, the decentralized security fabric consists of two sub-systems: (i) a security services layer based on the microservice-oriented architecture (MoA); (ii) a fundamental security networking infrastructure atop the Blockchain and DDS. To address heterogeneity and efficiency challenges such as developing and deploying security services in the distributed network environment, our security services layer adopts container technology to implement microservices for PC applications. The key operations and security schemes are decoupled into multiple containerized microservices. As container is loss-coupled from the 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 the SCs deployed on the Blockchain.

The Blockchain network acts as a decentralized and trust-free platform for security services, and it uses a scalable PoW consensus protocol to ensure the immutability and integrity of the on-chain data on the distributed ledger if the majority (51%) of the miners are honest. The security mechanisms are implemented by self-executing SCs, which are deployed on the Blockchain by trusted oracles such as system administrators. Thus, the security service layer can provide secure and autonomous microservices in a decentralized manner. To reduce the overheads of directly recording large data on the distributed ledger, we bring DDS into the security infrastructure as off-chain storage, which is built on a Swarm [26] network. Unlike the IPFS, which does not guarantee storage, Swarm maintains content-addressed DHT and relies on data redundancy to offer secure and robust data services. Moreover, the inclusion of incentives makes Swarm more flexible to integrate with the Ethereum Blockchain. The meta-data of point clouds and operation logs that require a heterogeneous format and various sizes are encrypted and then saved into the DDS. Raw data on the DDS can be easily addressed by their references (Swarm hash), which are recorded on the Blockchain for audition and verification. A Swarm hash has a 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 the transparent Blockchain.

4. Blockchain-Based Lightweight Point Cloud Data Authentication Framework

This section presents the 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 the participants and workflow in the framework. Then, we describe the structure of hybrid on-chain and off-chain storage. Finally, we explain the 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 the DDS. In this framework, owners can upload point clouds generated by 3D sensors to their private server, which acts as a service provider for the users of applications. By storing the access control policy and audit proof in the Blockchain, each owner can fully control its data, and the authorized user can verify the data stored on the private server. The overall workflow is divided into three stages according to the 3D point cloud life-cycle.

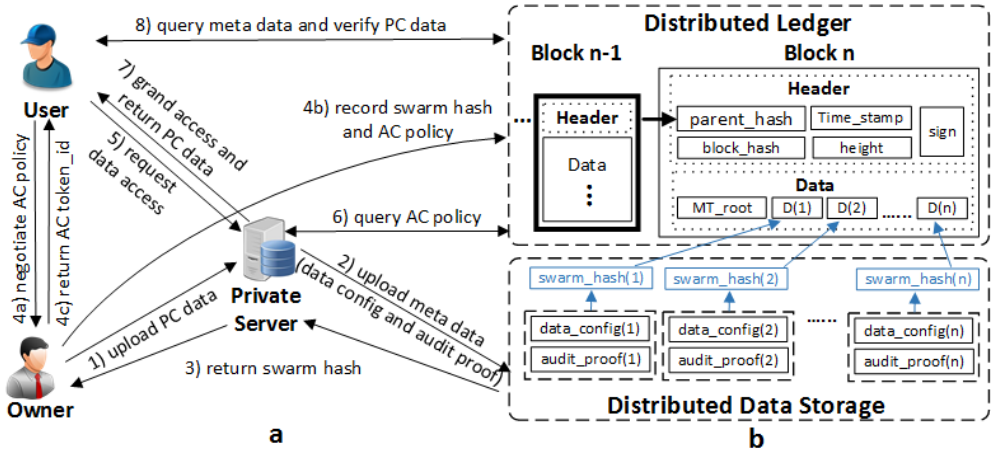

Figure 2. Illustration of Blockchain-based data authentication framework. (**a**) shows the workflow of 3D point cloud data storage, access authorization, and verification. (**b**) shows the structure of the hybrid on-chain and off-chain storage.

- *Data storage*: Owners and their private servers are in the same domain, and they can exchange secret keys via a trustworthy key distribution center (KDC). As a result, an owner and its private server can use shared secret keys to establish a secure communication channel for PC data transmission. In Step 1, the owner uses a shared secret key to encrypt point cloud data PC_i and then sends encrypted data to a private server. After receiving point clouds in Step 2, the private server stores encrypted PC_i into local storage and then records meta-data (e.g., configuration and audit proof) MD_i on the DDS. In a meta-data item, the configuration contains the URL address of a private server and other data properties such as the format and size, and the audit proof consists of an authenticator of raw data and a signature signed by the data owner. In Step 3, a site of the DDS stores the received MD_i and calculates a Swarm hash as a unique reference to address MD_i on the DDS. Finally, the Swarm hash is returned to the private server, and then, the private server transfers the Swarm hash to the data owner.
- *Data access control*: The data access control (AC) process is built on a capability-based access control (CapAC) scheme [27]. In Step 4a, a data user contacts a data owner to negotiate an AC policy for PC data sharing. Then, the data owner verifies the data user's identity and authorizes access rights for the data user given pre-defined AC policies. In Step 4b, the data owner stores the Swarm hash of the meta-data along with the assigned access rights in a distributed ledger (Blockchain). As long as the AC data have been successfully saved in an AC token on the Blockchain, a *token_id* is returned to the data owner. Finally, the data owner sends the *token_id* back to the data user as a notification, as Step 4c shows. In Step 5, a user first sends data access requests to a private server, which stores PC_i. Then, the private server retrieves the AC policy from the Blockchain and checks if the access rights assigned to the user are valid, as Step 6 shows. If the access authentication is successful, the private server uses shared secret keys to decrypt PC_i and return it to the data user, as Step 7 shows. Otherwise, the private server denies the access requests without sharing the data with unauthorized users.
- *Data verification*: To audit the received PC_i from a private server, the user queries the Swarm hash from the Blockchain and then retrieves meta-data MD_i from the DDB accordingly, as Step 8 shows. Because meta-data MD_i contains the audit proof that was submitted by the data owner when it uploaded PC_i, the data user can verify if PC_i satisfies the data properties and consistency of the authenticator and signature. In the data verification process, the user first checks if the properties of PC_i satisfy the configuration in MD_i. Then, it locally calculates the audit proof AP_i' according to

PC_i and compares it with the AP_i recorded in MD_i. If the audit proofs are equal, the data integrity has been guaranteed. Otherwise, the data may be inconsistent with the 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 a large data size and demand privacy preservation. Thus, it is impractical to directly store point clouds in a transparent Blockchain for data authentication. To ensure efficient and privacy-preserving data storage and sharing, we adopted a hybrid on-chain and off-chain storage structure in the data authentication framework, as shown in Figure 2b. In the point cloud data collection stage, the meta-data of the point cloud segments are saved in the DDS, while the raw data are managed by private servers. The meta-data MD_i contain the 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 the integrity of the authenticator of a point cloud segment and a signature singed by a data owner, which are byte strings with a small length. Therefore, the small size of the meta-data can greatly reduce the communication cost in the verification process. Furthermore, the meta-data are encrypted and then saved on the DDS, and only authorized users are allowed to query and decrypt the meta-data. It is promising to protect the privacy of data owners without exposing sensitive information on the Blockchain and DDS.

In our Swarm-based DDS, each of the stored meta-data has a unique Swarm hash as the addressable reference to the actual data storage, and any change of the stored data will lead to an inconsistent Swarm hash. Therefore, recording the Swarm hash on an immutable distributed ledger provides the non-tamperability property of the meta-data on the DDS. To verify the data integrity of a large point cloud file, the Swarm hash of meta-data MD_i is considered as a digest $D(i)$, which is located on a leaf of the 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 the sequential order will lead to different root hash values *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 the data verification process, a data user can query digests from the Blockchain and then in parallel validate the integrity of the segment data. Then, it can easily reconstruct the Merkle tree of the digests and obtain MT_root'. Finally, the data integrity of the entire point cloud file can be efficiently verified by comparing MT_root' with *MT_root* on the distributed ledger.

4.3. Decentralized Data Authentication Procedures

The 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 uploads them to the DDS, then appends the returned Swarm hash D_i to *ordered_swarm_hash*, as Lines 2–6 show. Following that, the 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, the data owner calls the 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 the data verification procedure, the data user firstly uses *token_id* as the input to call the smart contract function $query\_dataAC()$, which will return the public audit proof information stored on the Blockchain (Line 10). Regarding token validation, the data user performs function $BMT()$ on the received *ordered_swarm_hash* to recover the root hash mt_root', then checks if mt_root' is consistent with the audit proof *mt_root*. If the validation fails, it directly returns a false result. Otherwise, it goes ahead to the meta-data verification. Given the received *ordered_swarm_hash*, the data user traverses each digest D_i, which is used to download the meta-data MD_i from the DDS. Any wrong digest or corrupted

meta-data will lead to a $NULL$ result returned by the function $download\_data()$. Finally, a valid list of the meta-data is returned only if all meta-data can be successfully retrieved, as Lines 16–23 show.

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: $D_i \leftarrow$ **upload_data**(MD_i)
5: $ordered\_swarm\_hash$.append(D_i)
6: **end for**
7: $mt\_root \leftarrow$ **BMT**($ordered\_swarm\_hash$)
8: $receipt \leftarrow$ **Contract.set_dataAC**($token\_id, mt\_root, ordered\_swarm\_hash$)
9: **procedure**: verify_data($token\_id$)
10: $mt\_root, ordered\_swarm\_hash \leftarrow$ **Contract.query_dataAC**($token\_id$)
11: $mt\_root^{'} \leftarrow$ **BMT**($ordered\_swarm\_hash$)
12: **if** $mt\_root^{'} \neq mt\_root$ **then**
13: **return** $False$
14: **end if**
15: MD = []
16: **for** D_i in $ordered\_swarm\_hash$ **do**
17: $MD_i \leftarrow$ **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** $True, MD$

5. Experimental Results and Evaluation

In this section, the experimental configuration based on a proof-of-concept prototype implementation is described. Following that, we evaluate the performance of running SAUSA based on the numerical results, which is especially focused on the impact of the Blockchain on the system performance. In addition, a comparative evaluation of the previous works highlights the main contributions of SAUSA in terms of the lightweight Blockchain design, performance improvement, and security and privacy properties. Moreover, we analyze the security properties and discuss potential attacks.

5.1. Prototype Implementation

We used the Python language to implement a proof-of-concept prototype including client and server applications and microservices. A micro-framework called Flask [28] was used to develop RESTful APIs for the applications and microservices. We used standard python library cryptography [29] to develop all security primitives, such as the digital signature, symmetric cryptography (Fernet), and hash function (SHA-256). Solidity [30] was used for smart contracts' implementation and testing, and all SCs were 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's validation. A desktop emulated the private server, which stored the point clouds data managed by the data owner. To evaluate the impact of the hardware platforms on the data user side, both the Rpis and desktops were used to simulate a user client that requests data access. The private Ethereum network consisted of six miners, which are deployed on the cloud server as six containers separately, and each containerized miner was assigned one CPU core, while the other microservice containers that were deployed on the desktops and RPis worked

in light-node mode without mining blocks. All participants used Go-Ethereum [31] as the client application to interact with the smart contracts on the private Ethereum network. Regarding the Swarm-based DDS, we built a private Swarm test network consisting of five desktops as the service sites. Table 1 describes the devices that were used to build the experimental testbed.

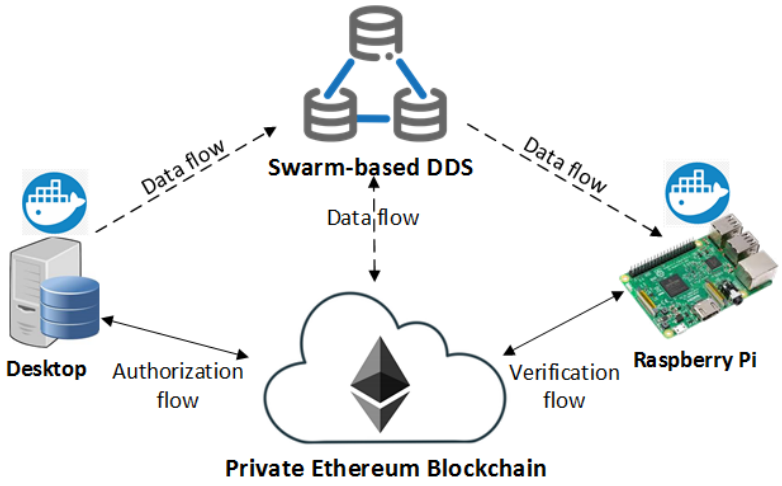

Figure 3. The experimental setup and network configuration.

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	500 GB HHD	64 GB (microSD)
OS	Ubuntu 20.04	Ubuntu 20.04	Raspbian (Jessie)

5.2. Performance Evaluation

This section evaluates the performance of executing the operations in the data authorization and verification. In the data authorization process, the desktop launches a transaction, which encapsulates the Swarm hash of the meta-data in the Blockchain, and then, the states of the SC can be updated until a block containing transactions committedby the miners. Thus, we evaluated the 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) the client (Rpi or desktop) queries the data token containing the Swarm hash of the meta-data and the root from the Blockchain; (2) the client validates the Merkle root and Swarm hash in the data token; (3) the client retrieves the meta-data from the DDS and verifies them. Therefore, we evaluated the processing time of the individual steps on different platforms by changing the number of meta-data (N_m). Finally, we analyzed the computational overheads incurred by retrieving the meta-data from the DDS and performing symmetric encryption on the meta-data. We conducted 50 Monte Carlo test runs for each test scenarios and used the averages to measure the results.

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

We scaled up N_m in the data authorization scenarios to evaluate how the size of the ordered list of digests (Swarm hash) impacts the performance. As a transaction's committed time is greatly influenced by the Blockchain confirmation time, we observed that all data authorization operations with different N_m demonstrated almost a similar end-

to-end latency (about 4 s) in our private Ethereum network. Regarding the computational complexity and processed data required by the SC, the gas used by the transactions may vary. Figure 4 shows the gas usage by data authorization transactions as N_m increases. The longer the ordered list of digests, the more gas is used per each transaction that stores the data on the Blockchain. Hence, recording the Swarm hash, rather than the meta-data or even the raw data on the distributed ledger, can greatly reduce the gas consumption of the Blockchain transaction.

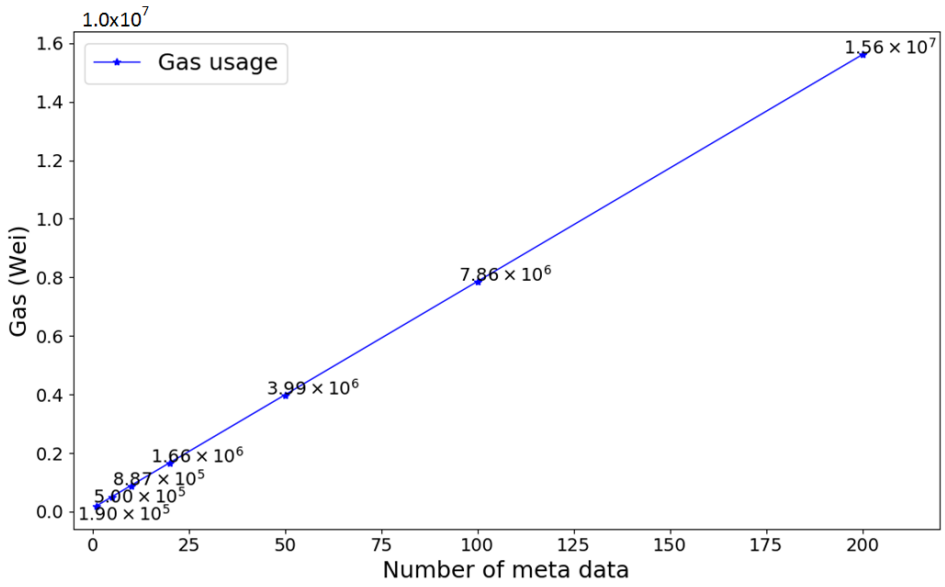

Figure 4. Gas usage in data authorization.

5.2.2. Processing Time by Data Verification

Figure 5 shows the average delays to evaluate how a data token query function of the 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 the SC needs more computational resources to process the data on the distributed ledger. Thus, the delays of querying a data token on both platforms scale linearly with N_m with the same gain. Due to different computational resources, the processing time of the data token query on the Rpis is almost double that on the desktops.

Figure 6 shows the computational overheads by validating token the data on the client side as N_m changes. The data token data validation requires reconstructing the binary Merkle tree of the ordered list of Swarm hashes, which results in a traversal complexity of $\mathcal{O}(N_m)$. Then, the root hash can be used as the fingerprint for all the meta-data to check for inconsistencies, which requires a computational complexity of $\mathcal{O}(1)$. Finally, the computational overheads incurred by verifying the token data are scale linearly with N_m. Computing the root hash of the binary Merkle tree demands intensive hash operations such that the computational power of the client machines dominates the performance of the data token validation. Therefore, a larger N_m in the data token validation brings more delays on the Rpis than the desktops. However, the impact was almost marginal in our test scenarios such that $N_m \leq 200$.

Figure 7 shows the processing time of verifying the meta-data on the client side as N_m increases. In the meta-data verification stage, a client uses the Swarm hash list in the data token to sequentially retrieve N_m meta-data from the DDS, which results in a communication complexity of $\mathcal{O}(N_m)$. Regarding the fixed bandwidth of the test network, increasing N_m allows for a larger round-trip time (RTT) and more computational resources in meta-data transmission. As a result, the delays of verifying a batch of meta-data are scale linearly with N_m. Unlike the desktops, the Rpis have limited computational resource to handle each data transmission. Therefore, the Rpis take a longer time to verify the same amount of meta-data than the desktops do.

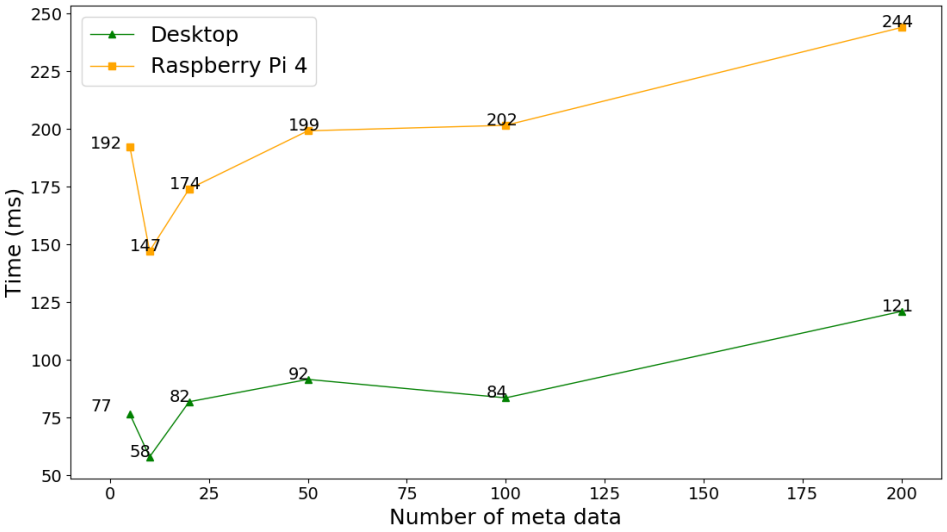

Figure 5. Latency by data token query on different platforms.

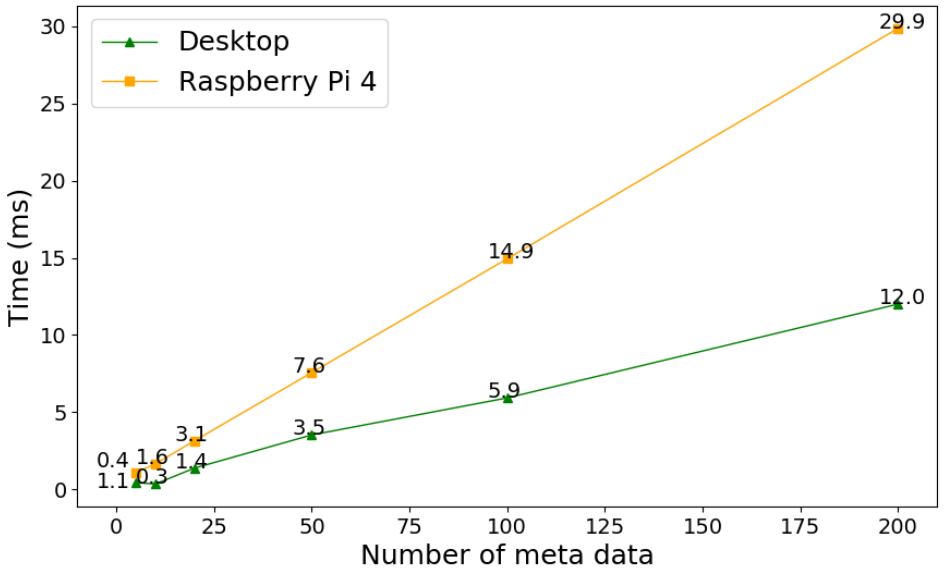

Figure 6. Processing time by data token validation on different platforms.

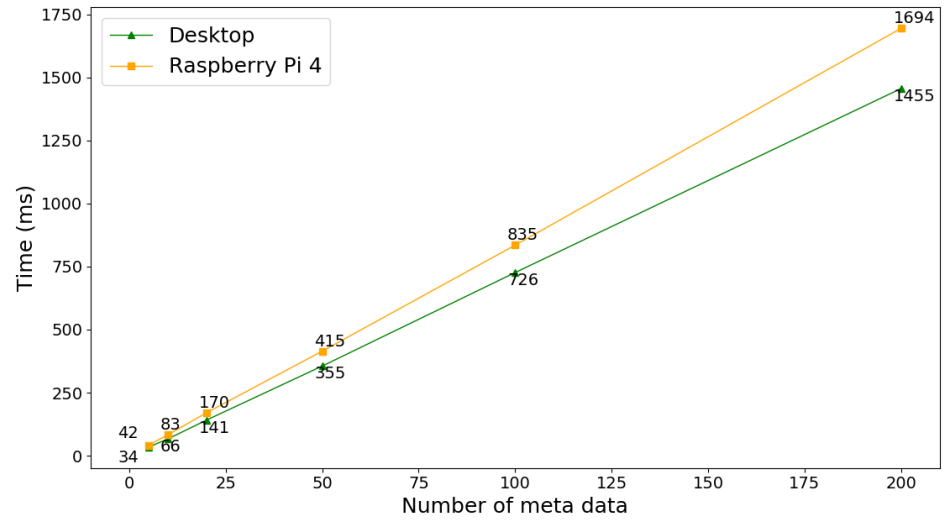

Figure 7. Processing time by meta-data verification on different platforms.

5.2.3. Computational Cost by Preserving Meta-Data Privacy

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

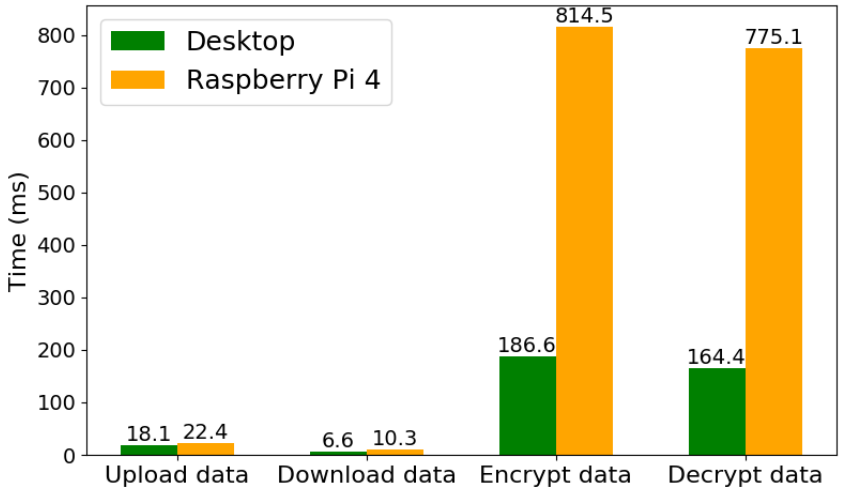

Figure 8. Processing time of meta-data operations: accessing Swarm and symmetric encryption.

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 security properties or implements some prototypes to evaluate system performance or other specifications. The symbol × indicates the opposite case. Unlike existing solutions, which lack details on the optimal network framework for QoS or evaluations on the impact of applying Blockchain to big data applications, we illustrate a comprehensive system architecture, along with details on the SDN-based service and lightweight data authentication framework. We especially evaluated the performance (e.g., network latency, processing time, and computational overheads) of the Blockchain-enabled security mechanism in the data access authentication and integrity verification process.

Table 2. Comparison among existing Blockchain-based solutions.

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

Regarding storage optimization and privacy preservation for point cloud data sharing, the hybrid on-chain and off-chain data storage structure not only reduces the communication and storage overheads by avoiding directly saving large volumes of raw data or audit proofs in Blockchain transactions, it also protects sensitive information by only exposing the references of encrypted meta-data on the transparent distributed ledger as the fingerprint proof. Unlike existing solutions, which 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 the robustness (availability and recoverability) of point cloud data sharing in multi-domain applications.

5.4. Security and Privacy Analysis

In this section, we first discuss the security and robustness of SAUSA and evaluate the impact of several common attacks on the proposed scheme. Then, we briefly describe the privacy preservation of SAUSA. Regarding the adversary model, we assumed that the capability of attackers is bounded by probabilistic polynomial time (PPT) such that they cannot compromise the basic cryptographic primitives, such as finding hash function collisions or breaking the cipher-text without knowing the secret keys. Moreover, we assumed that an adversary cannot control the majority of miners within the Ethereum network.

5.4.1. Sybil Attack

In a Sybil attack, an adversary can forge multiple fake identities to create malicious nodes. As a result, these malicious nodes can control the DDS network or even the consensus network to some extent. However, in the proposed SAUSA, permissioned network management provides the basic security primitives, such as the PKI and KDC for identity authentication and message encryption. Thus, all nodes with invalid identities are prevented from joining the domain networks. Furthermore, properly defined AC strategies are promising to reduce the impact of Sybil attacks across different application domains.

5.4.2. Collusion Tamper Attack

An adversary can compromise multiple nodes that collude to tamper with the PC data to influence the accuracy of 3D object detection and tracking. The collusion tamper attack could be easily achieved, especially for a small network. Our SAUSA anchors the meta-data of the original PC data to the Ethereum Blockchain. Once transactions encapsulating the meta-data are finalized on the immutable public distributed ledger, it is difficult for an adversary to attempt to revert the transactions or the status of smart contracts by controlling the majority (51%) of the nodes within a public Ethereum network. As the meta-data recorded on the Blockchain can be used as audit proofs for verifying the integrity of data on local private servers, the possibility of collusion tampering is reduced.

5.4.3. DDoS Attack

In conventional cloud-based systems, an adversary can accessmultiple compromised computers, such as using bots, to send huge volumes of TCP request traffic to target cloud servers in a short period of time. As a result, unexpected traffic jams by the DDoS attack overwhelm centralized servers such that service and networking functions become unavailable. Our solution adopts a DDS to achieve efficient and robust meta-data storage and distribution. As the DDS uses a DHT-based protocol to coordinate and maintain meta-data access service sites over a P2P network, it is hard for an adversary to disrupt the meta-data service by launching DDoS attacks to target service sites. Moreover, our data authentication framework relies on SCs deployed on Ethereum to ensure decentralization. Therefore, our approach can mitigate the impact of DDoS attacks better than centralized data auditing methods.

5.4.4. Privacy Preservation of PC Data

In data acquisition, users rely on trusted private servers to protect the raw PC data by AC policies and encryption algorithms. In the data sharing process, only encrypted meta-data along with references are exposed to the public network. The decentralized data authentication framework prevents attackers from violating access privileges or inspecting any sensitive information. However, the prototype of the SAUSA presented in this paper has no integrated privacy protection module to deter data privacy breach by honest or curious users, such as dishonest data users or private servers who attempt to obtain private information from PC data without deviating from pre-defined security protocols. Therefore, a data-privacy-preserving component based on differential privacy or secure multi-party computation is needed to guarantee PC data privacy, and we leave this for our future work.

6. Conclusions and Future Work

This paper presented 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 the QoS of point cloud applications. A lightweight data authentication framework atop the 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 demonstrated the effectiveness and efficiency of our SAUSA. However, there are open questions that need to be addressed before applying SAUSA to real-world 3D point cloud scenarios. We leave these limitations to our future works:

(1) SAUSA uses Ethereum to build a Blockchain network, which 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, such as Microchain [32], are promising to optimize computational utilization and improve performance in terms of end-to-end latency and transaction throughput. Our on-going efforts include validating SAUSA in a real-world point cloud scenario and the investigation of the integration of Microchain to reduce data authorization latency.
(2) This paper focused on the decentralized security scheme's 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.

Author Contributions: Conceptualization, R.X., Y.C., G.C. and E.B.; methodology, R.X. and Y.C.; software, R.X.; validation, R.X. and Y.C.; formal analysis, R.X. and Y.C.; funding acquisition, Y.C.; investigation, R.X. and Y.C.; resources, R.X. and G.C.; data curation, R.X.; writing—original draft preparation, R.X. and Y.C.; writing—review and editing, R.X. and Y.C.; visualization, R.X.; supervision, Y.C.; project administration, Y.C. All authors have read and agreed to the published version of the manuscript.

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

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

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
GDPR	General Data Protection Regulation
IoT	Internet of Things
IPFS	Interplanetary File System
KDC	Key distribution center
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 failure
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 Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [CrossRef] [PubMed]
2. Cao, C.; Preda, M.; Zaharia, T. 3D point cloud compression: A survey. In Proceedings of the 24th International Conference on 3D Web Technology, Los Angeles, CA, USA, 26–28 July 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), Orlando, FL, USA, 15–18 December 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 Netw.* **2016**, *30*, 58–65. [CrossRef]
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 Gener. Comput. Syst.* **2022**, *131*, 209–226. [CrossRef]
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), Kansas City, MO, USA, 16–19 September 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), Singapore, 11–13 December 2018; pp. 561–568.
8. De Deuge, M.; Quadros, A.; Hung, C.; Douillard, B. Unsupervised feature learning for classification of outdoor 3d scans. In Proceedings of the Australasian Conference on Robitics and Automation, Sydney, Australia, 2–4 December 2013; University of New South Wales: Kensington, Australia, 2013; Volume 2, p. 1.
9. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11621–11631.

10. Munoz, D.; Bagnell, J.A.; Vandapel, N.; Hebert, M. Contextual classification with functional max-margin markov networks. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 975–982.
11. Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A survey on software-defined networking. *IEEE Commun. Surv. Tutorials* **2014**, *17*, 27–51. [CrossRef]
12. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76. [CrossRef]
13. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: file:///C:/Users/MDPI/Downloads/21260-bitcoin-a-peer-to-peer-electronic-cash-system.pdf (accessed on 24 November 2022).
14. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst. (TOPLAS)* **1982**, *4*, 382–401. [CrossRef]
15. 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 Things J.* **2018**, *6*, 2188–2204. [CrossRef]
16. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**, *2*, 9. [CrossRef]
17. Liu, C.H.; Lin, Q.; Wen, S. Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning. *IEEE Trans. Ind. Inform.* **2018**, *15*, 3516–3526. [CrossRef]
18. 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 Trans. Parallel Distrib. Syst.* **2018**, *30*, 870–882. [CrossRef]
19. Yu, H.; Yang, Z.; Sinnott, R.O. Decentralized big data auditing for smart city environments leveraging Blockchain technology. *IEEE Access* **2018**, *7*, 6288–6296. [CrossRef]
20. 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. [CrossRef]
21. Li, H.; Han, D. EduRSS: A Blockchain-based educational records secure storage and sharing scheme. *IEEE Access* **2019**, *7*, 179273–179289. [CrossRef]
22. Ateniese, G.; Magri, B.; Venturi, D.; Andrade, E. Redactable Blockchain–or–rewriting history in bitcoin and friends. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, France, 26–28 April 2017; pp. 111–126.
23. Krawczyk, H.; Rabin, T. Chameleon Hashing and Signatures. 1998. Available online: https://eprint.iacr.org/1998/010 (accessed on 24 November 2022).
24. Politou, E.; Alepis, E.; Patsakis, C.; Casino, F.; Alazab, M. Delegated content erasure in IPFS. *Future Gener. Comput. Syst.* **2020**, *112*, 956–964. [CrossRef]
25. Campanile, L.; Cantiello, P.; Iacono, M.; Marulli, F.; Mastroianni, M. Risk Analysis of a GDPR-Compliant Deletion Technique for Consortium Blockchains Based on Pseudonymization. In Proceedings of the International Conference on Computational Science and Its Applications, Cagliari, Italy, 13–16 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 3–14.
26. Swarm. Available online: https://ethersphere.github.io/Swarm-home/ (accessed on 30 September 2022).
27. 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. [CrossRef]
28. Flask: A Pyhon Microframework. Available online: https://flask.palletsprojects.com/ (accessed on 30 September 2022).
29. Pyca/Cryptography Documentation. Available online: https://cryptography.io/ (accessed on 30 September 2022).
30. Solidity. Available online: https://docs.soliditylang.org/en/v0.8.13/ (accessed on 30 September 2022).
31. Go-Ethereum. Available online: https://ethereum.github.io/go-ethereum/ (accessed on 30 September 2022).
32. Xu, R.; Chen, Y.; Blasch, E. Microchain: A Light Hierarchical Consensus Protocol for IoT Systems. In *Blockchain Applications in IoT Ecosystem*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 129–149.