applied sciences

MDPI

Article

BDIDA-IoT: A Blockchain-Based Decentralized Identity Architecture Enhances the Efficiency of IoT Data Flow

Zequan Yang [1,†], Yumeng Liu [1,†], Xiaopeng Jin [1,*], Xiaoling Luo [2,*], Yuan Xu [1], Meng Li [1], Peng Chen [1], Bixia Tang [1] and Baohui Lin [1]

[1] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China; 2110416004@stumail.sztu.edu.cn (Z.Y.); liuyumeng@sztu.edu.cn (Y.L.); xuyuan2@sztu.edu.cn (Y.X.); limeng2@sztu.edu.cn (M.L.); 201902010214@stumail.sztu.edu.cn (P.C.); 2310413021@email.szu.edu.cn (B.T.); 202100203020@stumail.sztu.edu.cn (B.L.)

[2] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518061, China

* Correspondence: jinxiaopeng@sztu.edu.cn (X.J.); xlluo@szu.edu.cn (X.L.); Tel.: +86-176-0305-7669 (X.J.); +86-156-2543-0446 (X.L.)

† These authors contributed equally to this work.

Abstract: With the booming development of the Internet of Things (IoT) industry, millions of data are generated every day. How to use and manage these data safely and efficiently has become a hot issue of concern to people. Due to the accumulation of IoT data, the isolated data island phenomenon makes it difficult to connect and interact with each data owner, and the security and privacy of IoT data also become a challenge. Blockchain is a decentralized database technology that uses distributed accounting to ensure reliable data transmission and access, along with smart contracts that can be executed automatically to program and manipulate data. At the same time, blockchain techniques have stronger security and privacy, which can better meet the needs of users. In this paper, we analyze the current management mode and challenges of IoT data and propose an IoT data platform based on blockchain. The proposed platform takes into account important factors including IoT data traceability, IoT data trusted transactions, etc. Our IoT data platform achieves the trusted management and transaction of IoT data. We also propose follow-up optimization solutions to expand the application scope of our platform and achieve more efficient management of IoT data.

Keywords: internet of things (IoT); blockchain; decentralized identifiers (DIDs); trusted transactions; data privacy; data management

Citation: Yang, Z.; Liu, Y.; Xiao, P.; Luo, X.; Xu, Y.; Li, M.; Chen, P.; Tang, B.; Lin, B. BDIDA-IoT: A Blockchain-Based Decentralized Identity Architecture Enhances the Efficiency of IoT Data Flow. *Appl. Sci.* **2024**, *14*, 1807. https://doi.org/10.3390/app14051807

Academic Editor: Gianluca Lax

Received: 19 January 2024
Revised: 17 February 2024
Accepted: 20 February 2024
Published: 22 February 2024

1. Introduction

With the advent of the IoT era, the IoT era will change. The IoT is a connected environment where every device can communicate seamlessly, and various devices will be able to participate in different communication channels. The data emitted by each IoT device will move beyond mere raw data into personalized insights tailored to user preferences and, in some cases, aggregated with other data. The basic concept of the Internet of Things is very simple, but its widespread application is expected to spark innovation and push traditional technologies forward.

In contrast to the pre-IoT era, where users primarily relied on data provided by service providers, the advent of the IoT grants users direct access to sensors. This enables users to send instructions directly to applications, facilitating seamless and relevant operations. The data harnessed from the IoT not only transforms the user experience but also serves as the foundation for novel services catering to industries, academia, and individuals alike [1].

Currently, popular approaches involve utilizing various database technologies, including distributed database technologies, for effective data management. However, a prevalent challenge arises as data owners often specialize within specific industries. As enterprises grow and diversify into multiple business divisions, each division accumulates its distinct

dataset. Unfortunately, these datasets are frequently siloed, stored separately, and governed by unique definitions. The resulting landscape resembles isolated data islands, hindering the seamless connection and interaction of data across different divisions within an enterprise [2]. This phenomenon, termed isolated data islands, underscores the imperative to establish an efficient and secure data management paradigm to facilitate seamless collaboration among diverse data owners in the IoT.

As a new type of system, blockchain technology is expected to change the organization of IoT systems. Unlike traditional methods of routing data through a central processing unit, blockchain provides decentralized point-to-point connections for seamless data transfer. This decentralization empowers distributed computing to handle a staggering volume of transactions, reaching into the realm of billions [3]. Simultaneously, the latent computing power, storage capacity, and bandwidth residing in millions of idle devices dispersed across various locations can be harnessed to their full potential. This utilization of idle resources contributes significantly to the processing of transactions while substantially reducing computing and storage costs [4].

The combination of blockchain technology with smart contracts is a good application that turns every smart device into an autonomous network node. These nodes execute predefined or embedded rules, facilitating functions such as information exchange and identity verification with other nodes. This innovative approach ensures that IoT products remain relevant and functional throughout their life cycle, minimizing equipment maintenance costs and mitigating the risk of obsolescence [5].

The application of blockchain technology provides a way of thinking about the problems that exist in the Internet of Things, diminishing or eliminating the need for third-party authentication. It offers a clear solution to scalability, single points of failure, time stamping, logging, privacy, trust, and reliability concerns. Through the utilization of smart contracts, IoT devices can participate in secure message exchange, simulating agreements between parties without centralized authorization. The fusion of blockchain and the IoT has prompted extensive research efforts. For instance, Kumar et al. [6] have proposed scalable blockchain frameworks, ensuring data integrity and secure transmission, with the added security layer of Ethereum smart contracts. BCoT (Blockchain for IoT), introduced by Banerjee et al. [7], delineates an architecture that amalgamates blockchain characteristics with IoT, outlining promising application prospects. Azbeg et al. [8] have delved into designing a secure medical system, addressing concerns such as security, scalability, and processing time. Their solution incorporates data hashing, smart contracts, and the Inter-Planetary File System (IPFS) to ensure data security and credibility.

Blockchain technology can provide inherent anonymity to all parties to a transaction; in other words, the nodes in the blockchain are all peers. The use of hash-value addresses as unique identifiers on the blockchain, while ensuring privacy, may pose challenges in facilitating seamless data flow between these parties. Decentralized Identifiers (DIDs) have emerged as a crucial solution to address identity concerns within decentralized blockchain systems [9]. DIDs play a pivotal role in minimizing the risk of user credential exposure by furnishing relevant contextual information based on the specific information that needs to be disclosed. In our innovative solution, DIDs are employed to establish and verify both user and data identities.

In order to offer greater flexibility, users have the option to deploy smart contracts for the transaction of data rights and reduce the need for direct data delivery. This forward-thinking approach, facilitated by federated learning [10], introduces a dynamic paradigm for the exchange of data rights. The blockchain serves as a comprehensive repository, where all transaction and storage information can be seamlessly queried, contributing to the overall traceability of IoT data [11]. This not only fortifies transparency but also establishes a foundation for building trust within the intricate ecosystem of IoT data transactions.

Building upon prior research findings, we present an innovative framework dedicated to the secure management and transaction of IoT data based on blockchain technology. Our platform is designed to capitalize on the distinctive features of blockchain, creating an IoT

data trading system that facilitates transactions involving data sourced from the IoT devices and sensors. Our system empowers individuals to acquire data from diverse IoT sources, ensuring both the secure transmission of data and the execution of payments through peer-to-peer (P2P) transactions. The proposed system is able to allow users to upload summaries of data to the blockchain, providing them with a way to generate revenue. According to the relevant mechanisms of blockchain technology, the integrity of transaction data is guaranteed, and trust and reliability are promoted. The addition of smart contracts further improves the efficiency of transactions on the blockchain, simplifying the process and enhancing the user experience.

The following sections of this article are as follows: First, we introduce the technologies involved in the system we designed, including analyzing the inherent advantages and disadvantages of these technologies to understand their implications. Next, we explained our system design, including the overall architecture, the blockchain composite layer, and the complex data transaction process. Following that, the practical implementation of our proposed system is introduced, along with the testing of relevant parameters. It also includes discussion and analysis of the survey results. Finally, a summary of this paper is provided, along with suggestions for future research. This systematic approach ensures a structured and informative exploration of our technical efforts, system design, implementation, and future avenues of exploration.

2. Preliminaries

2.1. Bitcoin Structure

Our blockchain system has been conceptualized based on the fundamental principles inherent in the architecture of Bitcoin [12]. Drawing inspiration from the Bitcoin network, our system offers the inclusivity of any participant within the expansive blockchain network. Each node takes charge of up-keeping a local ledger and fortifying accounting rights through robust consensus mechanisms. In alignment with the spirit of Bitcoin, our system is founded on principles of openness, decentralization, transparency, traceability, and immutability. These key attributes collectively position our platform as an optimal solution for safeguarding critical information pertaining to various circulation processes.

Just as in the Bitcoin network, our blockchain system accommodates a diverse range of participants, fostering a decentralized and transparent ecosystem. This inclusivity ensures that any entity, be it an individual or organization, can actively participate in the network, contributing to the collective maintenance of the blockchain [13]. The local ledger maintained by each node serves as a repository for transactional data, ensuring that the entire network is synchronized and traceable.

Moreover, the consensus mechanisms embedded play a pivotal role in establishing trust and legitimacy among network participants. Through a distributed agreement on the validity of transactions, blockchain ensures the immutability of recorded data, making it resistant to tampering or unauthorized alterations [14]. This robustness enhances the security and reliability of the information stored within the blockchain, creating a trustworthy foundation for critical circulation processes.

Figure 1 presents a concise organizational chart illustrating the structural framework of the Bitcoin system.The interconnected nature of blockchain ensures that transactions are seamlessly chained, facilitating easy retrieval of transaction origins and destinations. The blockchain is a chain structure composed of this transaction information. At the same time, transactions are also the main content stored in the blockchain nodes. The transaction structure encompasses metadata, as well as input and output information related to the transaction. The overall structure of Bitcoin is chain-like, so historical records can be traced bit by bit. In the transaction body, the most important thing is the transaction field, which stores specific transaction information and needs to ensure the query of its content, etc. This data typically represents the transactions conducted, serving as a virtual ledger that traces the input and output redirection of transactions. This simplicity effectively prevents undue expansion of the overall blockchain system size.

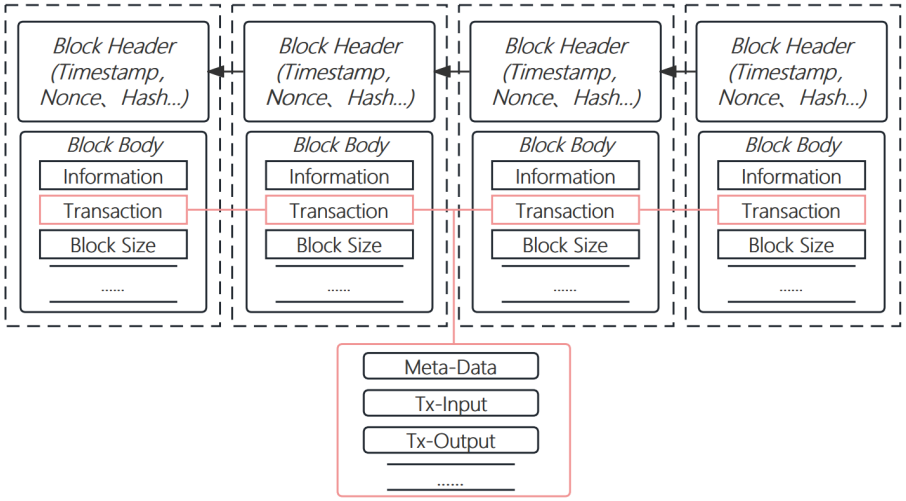


Figure 1. Bitcoin structure diagram. The block header stores some verification information of this block, and the block body stores the specific information of this block, of which transaction information is the main storage part. The pink box in the figure represents transaction information.

2.2. Decentralized Identifiers (DIDs)

In the Bitcoin system, transaction source and destination are simply recorded, ensuring that identity information remains non-sensitive for any participant in the blockchain. However, in a decentralized system such as blockchain it is necessary to establish a trusted identity mechanism. Especially when applied to specific scenarios, the need arises to authenticate transaction data or representatives based on identity. In such cases, user identity information becomes essential.

Decentralized identity [15] is an identity mechanism adapted to a decentralized environment. It constitutes a standard for identifying entities, including individuals, organizations, or devices. DIDs, or digital identity identifiers, uniquely distinguish entities without reliance on centralized identity providers. The primary objective is to overcome the limitations of traditional identity verification models, empowering individuals to manage and control their digital identities effectively.

As depicted in Figure 2, DIDs adopt the Uniform Resource Identifier (URI) format, commencing with "did:" and followed by a unique identifier, such as "did:example:123456". This identifier is linked to a DID Document presented in JSON format, containing DID-related details such as public keys, authentication services, and endpoints. This structure enables others to verify DID ownership and employ the DID's public key for authentication or data encryption. Due to the diversity of DID information, different users may choose different verification methods during DID verification.

The DID Document encodes public key information in the form of a JSON Web Key (JWK). Different DIDs may utilize distinct encryption algorithms and key types, resulting in potential variations in verification steps depending on the specific DID implementation [16].

DIDs feature decentralized management, realized through the adoption of DID methods. DID methods constitute a set of specifications defining how DID identifiers are created, parsed, and updated. These methods use different technologies and protocols to cater to diverse application scenarios. For instance, Ethereum's smart contracts can implement the related DID verification process [17]. Fotiou et al. [18] utilize DIDs to construct a decentralized identity identification system, addressing identity verification and authorization challenges in content-centralized networks. Fan et al. [19] enhance the security and credibility of IoT networks by introducing decentralized identity and access management mechanisms. Zhu et al. [20] propose an identity management system based on range proof that provides an efficient, privacy-secure authentication solution to the social networking field, providing valuable ideas for the research and practical application of decentralized identity management.

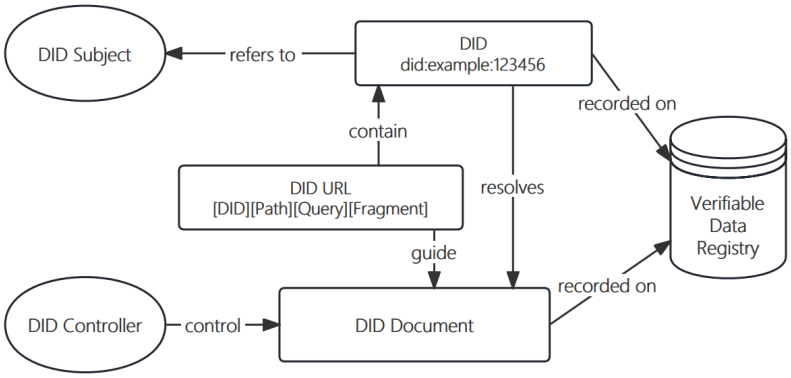


Figure 2. Organizational structure of DIDs. A DID Document is a collection of identity information that is invisible to the outsider and is claimed by the DID Controller field. The DID-ID part is the unique identifier of the DID Document, from which the specific DID Subject can be located. The DID URL is the external service provided by the DID, including authentication. The Verifiable Data Registry is the agency responsible for issuing verifiable credential.

The integration of DIDs introduces enhanced flexibility, privacy protection, and user control to the digital identity domain. Users gain the capability to share and verify identities across various networks and applications without relying on centralized authentication services. This advancement contributes to the creation of a more secure, transparent, and user-friendly digital identity ecosystem, thereby promoting widespread adoption and application [21].

In the BDIDA-IoT solution, DID is used to uniquely identify a certain piece of data or a user, which means that the controller that identifies the DID of the data can be another DID that identifies the user. DID is not static data, it is updated dynamically, and the update status of DID can be recorded in the blockchain immutably.

2.3. Elliptic Curve Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) [22] stands as a widely employed digital signature algorithm within the realm of public key cryptography and is renowned for its robust security and efficient performance. In the elliptic curve algorithm, two fundamental operations, namely Point Addition and Point Multiplication, play a pivotal role.

Consider an elliptic curve described by the equation $y^2 = x^3 - 7x + 9$. Drawing a straight line intersecting the curve at three distinct points—P, Q, and R—becomes a pivotal illustration. According to the definition of the point addition operation, the sum $P + Q + R$ equals the identity 0, implying $P + Q = -R$. The negation of R is defined as a point obtained through symmetry about the x-axis, as depicted in Figure 3, encapsulating the essence of point addition.

If we adjust the position of this straight line such that points P and Q coincide, adhering to the aforementioned point addition rules, we can derive the point $2P$. Following the same principle, by consecutively connecting point P to nP, we obtain $3P$, $4P$, and so forth, up to $(n + 1)P$. The dot product is denoted as $K \times P$, signifying K iterations of the point addition operation for point P.

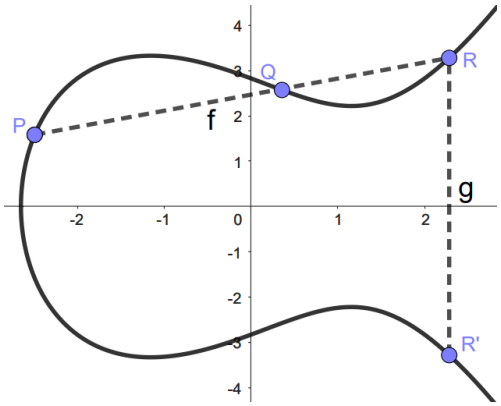


Figure 3. Representation of point addition diagram on a curve. Points P and Q are selected points, and Point R can be obtained by intersecting the PQ line and the curve. Then, point R is perpendicular to the x-axis to obtain point R', which means $P + Q = R'$ point summation form.

2.3.1. Security, Encryption, and Decryption Performance

ECDSA is founded upon the discrete logarithm problem on elliptic curves, leveraging the mathematical complexity of this problem to provide formidable security. In comparison to traditional algorithms such as the Rivest–Shamir–Adlema (RSA) algorithm, ECDSA achieves significant security strength, even with shorter key lengths. This characteristic renders it particularly suitable for resource-constrained environments such as IoT devices and mobile applications.

ECDSA exhibits notable advantages in encryption and decryption performance when compared to traditional algorithms. The efficiency of elliptic curve operations, as opposed to integer factorization, allows ECDSA to deliver substantial security with relatively short key lengths, thus easing the challenges associated with key management and computational resources. This efficiency positions ECDSA favorably in scenarios with heightened computing performance demands, including wireless sensor networks, smart cards, and mobile devices.

2.3.2. Signature Verification Process

1. Generate Key Pair: Key generation involves selecting a private key and calculating the corresponding public key, $P0$, using the base point $G0$ on the elliptic curve and the private key, $K0$. The calculation follows the formula:

$$P0 = K0 \times G0 \tag{1}$$

2. Signature Process: The signature process entails selecting a random number, $k1$, calculating the corresponding elliptic curve point, $P1$, generating the x-coordinate, R, of the elliptic curve point as a component of the signature, and ultimately obtaining the signature through a series of calculations. The formula involves a hash value H obtained through a hashing algorithm applied to the message to be signed and a random number p.

$$S = k1^{-1}(H + K0 \times R) \bmod p \tag{2}$$

While the signature process of the elliptic curve algorithm may be somewhat more intricate than other signature algorithms, this complexity does not impede its widespread application. Optimization and hardware support can enhance the efficiency of mathematical operations on elliptic curves, mitigating any perceived complexity.

3. Verification Process: The verification process involves calculating and comparing points on the elliptic curve. It ensures the integrity and authenticity of information by verifying the legitimacy of the signature. The public key $P0$ is utilized in this process,

and verification is successful if the x-coordinate of P matches R. Importantly, the private key is not involved in the signature verification process.

$$P = S^{-1} \times H \times G + S^{-1} \times R \times P0 \quad (3)$$

The elliptic curve algorithm is widely used in digital signatures because of its simple signing and verification process. This algorithm has good performance and its security has been verified in long-term use. It is one of the most widely used signature algorithms. In the BDIDA-IoT solution, we use the elliptic curve algorithm to verify the Verifiable Credential issued by the DID. During the verification process, users can choose to use different types of elliptic curve algorithms, and the overall verification process is automated.

3. Proposed Hybrid Framework

Existing frameworks are mainly divided into two categories: Public-Chains (PB), based on Ethereum and others, and solutions based on Consortium Blockchain (CB). Our solution is a modification of the blockchain structure and content based on reference to the Bitcoin source code, which can be well adapted to the application scenarios of Internet of Things data circulation. As stated in Table 1, In terms of transaction speed, BDIDA-IoT can appropriately reduce the difficulty of reaching consensus through the trust foundation of decentralized identity. In terms of supervision, the behavior of a user can be traced based on the decentralized DID. In terms of identity management, BDIDA-IoT combines the mechanism of decentralized identity, while PB requires additional compatibility.

Table 1. Comparison between BDIDA-IoT and public chains and consortium chains.

	PB	BDIDA-IoT	CB
Transaction speed	Slow	Moderate	Fast
Type of transaction	Direct (Digital Wallet)	Direct (Digital Wallet)	Proxy
Regulatory	Weak	Moderate	Strong
Openness	Strong	Strong	Weak
Credibility	Strong	Strong	General
Identity management	Optional	Decentralization	Centralization

BDIDA-IoT is a blockchain platform that references the source code of Bitcoin. Ethereum's transactions require the additional cost of Gas, while the associated features deployed using smart contracts lack good scalability. In our solution, we can add different plug-ins to support more services according to the needs. Related functions can be opened in the form of apis in the future to support more development needs.

In terms of trust mode, PB allows any user to access the blockchain network and ensures the trustworthiness of the block through a high degree of difficulty. CB requires users to register with a third party to demonstrate the trustworthiness of their identity. BDIDA-IoT uses the DID scheme to map users and data into concrete DID Documents. The transaction information in the blockchain is replaced with a change in DID permissions, which means the nodes in the blockchain do not need to care about the identity information of other users, thus reducing the block time. In terms of data flow, PB and CB cannot directly identify the change of permission of data. BDIDA-IoT realizes the fine-grained permission changes of data through the unique identification pattern of DID and the addition of related data structures.

Issues such as data tampering, unauthorized access, and data privacy exist in traditional IoT architectures. In terms of data tampering, IoT data need to calculate a data digest and add it to the DID Document before it is uploaded to the chain. During the data transfer process, we can determine whether the data has been tampered with by checking whether the data summary is consistent before and after, thereby ensuring data consistency. In terms of unauthorized access, we can check the controller field in the DID Document to determine whether the user has relevant permissions for the data, thereby solving the

data permission problem. In terms of privacy protection, the original data are not stored on the blockchain. Users can query the entire life cycle of the data they own through the blockchain, and they can realize their privacy concerns about the data by controlling the DID Document.

3.1. Framework Description

The proposed blockchain system framework in this study is shown in Figure 4. Based on the traditional three-tier structure of the IoT, the framework adds a blockchain composite layer. The addition of the blockchain composite layer adds a buffer between the application layer and the transport layer and relieves the pressure of the application layer when facing massive data delivery. The perception layer, at the bottom of the framework, consists of IoT devices large and small and is a key part of information collection. The transport layer is a network cluster with seamless connection and all-round coverage of the IoT, and its main function is to transmit information acquired by the perception layer. The network transmission function of the transport layer overlaps with the network layer in the blockchain composite layer, which is set as a separate layer for convenient representation. The function of the application layer is information processing. The application layer, together with the lowest sensing layer, is the salient feature and core of the IoT. The application layer can calculate, process, and mine the data collected by the sensing layer, so as to realize the real-time control, accurate management, and scientific decision-making of the physical world.

In the proposed hybrid framework, IoT data can be transmitted to the sensing layer via IoT protocols and then be converted into the appropriate blockchain transaction format and transmitted to other nodes through the blockchain network. Blockchain nodes can verify the legitimacy of transactions and add them to the blockchain to ensure the security and immutability of the data. This combination can provide greater security, trust, and transparency to IoT systems, while enabling data exchange and sharing across devices and platforms.

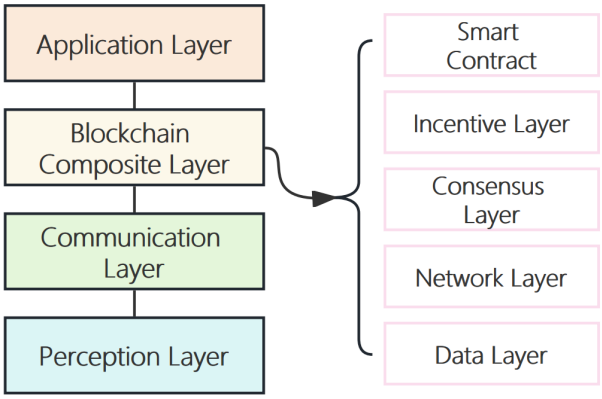


Figure 4. Schematic diagram of the overall framework. We added the Blockchain Composite layer to the traditional IoT architecture, which is subdivided into a 5-layer structure.

In our framework, when more and more IoT devices start to connect to the system, the result is more and more data transaction information. In our design, a block can store up to 4000 transactions, each transaction can declare 500 MB of data, and each block can be up to 1 MB. This means that 1 GB of space can claim up to 1000 blocks and 500 GB of data, so our framework is able to guarantee scalability of the ability to handle an increasing number of IoT devices and transactions. It is worth mentioning that as more and more IoT devices begin to connect to the blockchain network, the processing power and stability of our system will also increase.

In the blockchain composite layer, the data layer defines the data structure of the blockchain and the chain and formulates relevant standards for various information in the blockchain, such as the storage of blockchain information, Unspent Transaction Output

(UTXO), and Data Stored Sets (DSSs). The network layer is responsible for the access and verification of blockchain nodes (users). The consensus layer is responsible for synchronizing local blockchain, UTXO, and DSS information between nodes; verifying the legality of each packaged block; and maintaining the operation sequence and fairness of the system. The incentive layer sets up relevant rewards for users who participate in maintaining the blockchain. A smart contract is a piece of code deployed on a blockchain, where users can deploy the corresponding smart contract to realize data-related functions. It differs from the application layer in that it is a piece of code on a blockchain and can provide lightweight and decentralized services.

The flow of data is based on DIDs. A DID should be issued by one or more trusted third parties, and this DID can be verified by other users and provide relevant verification services. In the blockchain system, users can be represented by public key information, which uniquely identifies a user, and this public key information is also the identity identifier of the DID decentralized identity. In Figure 5, a DID Document should include the Context, ID, Controller, Public-Key, Service, and Authentication fields. The content of Context is the version information of the DID protocol and some explanations about the DID, such as data size, data type, data organization form, usage method, etc. ID is the number of the DID, and a hash value is generated according to the content of the DID Document to represent it. Controller is the owner of the DID Document, and this field can have multiple users. Public-Key is a collection of public keys used to verify a user's ownership of the DID. Service is the relevant service information provided by the issuer of the DID, and the content is a URL. Authentication is the organization that issues the DID, and this field can be a collection. In the process of data flow, the fields of a DID Document corresponding to a set of data should be changed at any time. The Controller and Public-Key fields need to be added or modified at any time. The changes are saved on the blockchain, and each change needs to generate a data digest pointing to the historical change record of the DID, so that we can perform relevant traceability operations.

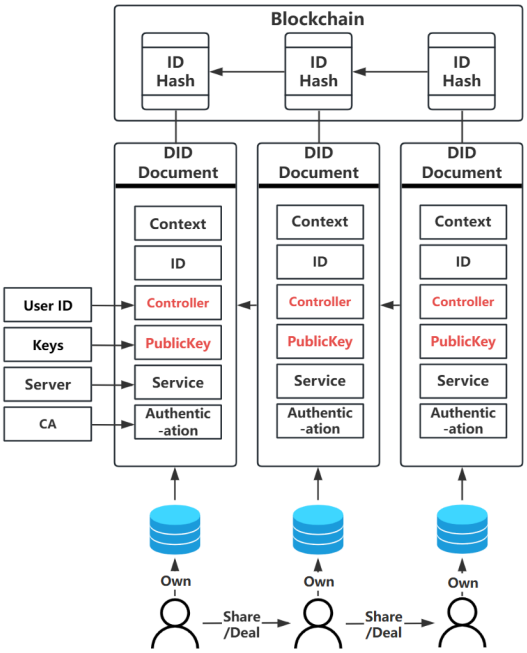


Figure 5. Data flow process based on DIDs. The change of data ownership is accomplished by modifying the 'controller' field in the Decentralized Identifier (DID) associated with a specific data segment. This modification process is recorded on the blockchain, ensuring that the entire life-cycle of the data is traceable. The red part indicates the ownership and public key of the DID, which is an important attribute of the DID.

3.2. Blockchain Composite Layer

The user's data transaction process is realized at the blockchain composite layer. Users join our platform by connecting to the blockchain network. Whether or not a user is allowed to access the blockchain network was included in a study by Steichen et al. [23]. Steichen et al., utilized smart contracts to maintain access control lists (ACLs), which enables network access control for users. In the blockchain network, users can earn coins by providing data and mining, and coins are used in our system to measure the trustworthiness of a use and also to reflect the user's activity. There are two types of transactions in the transaction pool: storage transactions and data transactions. Users declare their data ownership to other users through storage transactions. Data transaction corresponds to the act of buying and selling data between users. Meanwhile, data transactions correspond to changes in UTXO, while storage transactions correspond to changes in DSS. We set the DSS to speed up the querying of the amount of data and smart contracts owned by users so that we do not have to waste any significant time traversing the blockchain.

3.3. Data Stored Set

In addition to maintaining UTXO locally, nodes added to blockchain also need to maintain a data store set DSS, as shown in Figure 6. The purpose of our data store set is to speed up the query of the amount of data on the node chain. Instead of traversing the entire blockchain and spending significant time, users can just look up information on the data store set.

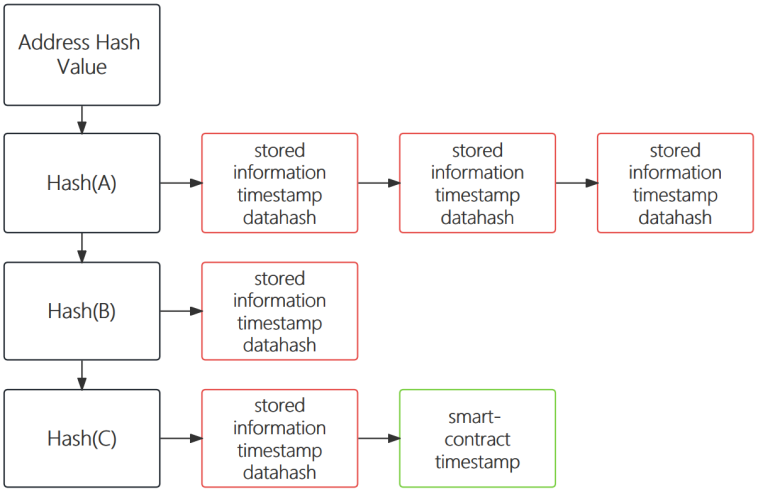


Figure 6. The structure of DSS, including hash-head node and two types of transaction. The two types of transaction after the hash header node have different meanings. Generally speaking, storage transactions are the majority, while contract transactions are the minority. The data declared by users through the blockchain is represented by storage transactions, while only one contract transaction can handle multiple data.

The red transaction in the picture represents a storage transaction. Each storage transaction indicates that a certain address has declared 100 MB of data in the blockchain. The green transaction in the figure represents a contract transaction, which is the public information after the user deploys the smart contract. Other users can obtain the information of calling the smart contract by querying the contract transaction, which can better improve the efficiency of the transaction.

DSS is a collection of hash tables that can be looked up by setting address–value pairs. When a user successfully publishes a data store transaction, each node in the blockchain needs to add the user's address to the data store and add related records. Values are chained structures, where each block record stores information, time, data hash, and contract address. We stipulate that each stored transaction of the user can only declare 100 MB of data at a time, and 100 MB of data will generate a random number of 32 B after

the data summarization algorithm. A block of the blockchain is 1 MB in size, with 80 B block headers and variable block bodies. The transaction information is stored in the block body, which means that a block can contain up to 2.5 GB of data storage information, thus preventing the blockchain from being too long. If a user successfully publishes a storage message, the user will locate the message through the address first in DSS. If there is no record of the address, a new record will be added. If it exists, the chain structure is traversed in the value corresponding to the address. Each block represents 100 MB of data for the address to be linked. The process of querying user data volume is to traverse the chain structure of an address.

3.4. Transaction Process

Figure 7 is an overview of the proposed transaction process, which is mainly composed of various edge computing devices, transaction pools, and blockchain. By default, all nodes have successfully joined the blockchain network. The nodes in Figure 7 represent individual user entities with data; miners can be entities that do not own data but have enough computing power, or they can be entities that own data. They are the main workers to maintain the normal operation of blockchain. Miners can receive corresponding rewards through mining. The trading pool stores the trading information or storage information published by the node; blockchain stores information about activity between nodes.

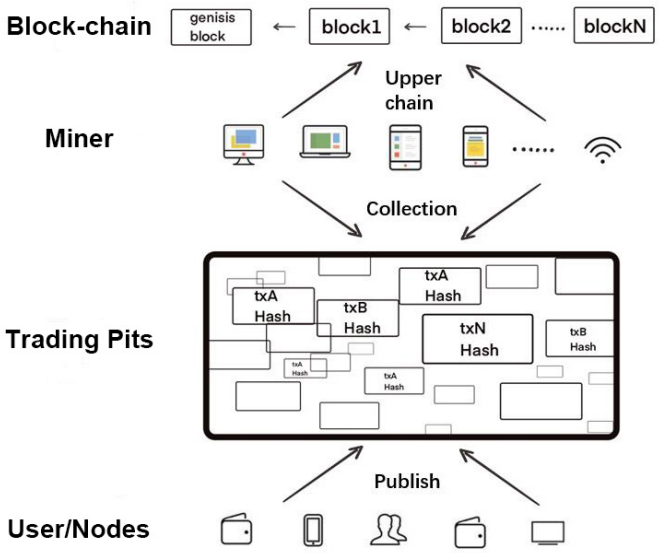


Figure 7. Overview of transaction Process. The transaction process varies depending on the role of the node. The role of a node in the system can be multiple at the same time. It can be a node that initiates data transactions or a node that declares data. Depending on the role, the work required by the node is also different, but the work performed by a single node is determined based on specific needs.

If node A, with a large amount of IoT data, can provide data for other nodes to use in order to receive a reward, node A first needs to publish a stored transaction to the transaction pool. The stored transaction includes the hash value of the data, the timestamp, and node A's address. Miners select the transaction from the trade pool and package it into a block, which can be uploaded to the blockchain after other miners verify that the block is valid. When other nodes need to use the data, the blockchain can be queried to see which nodes have the data in order to initiate transactions.

If node A needs to use the local data of node B, the process is as follows: The node searches the DSS data storage set to see if node B has enough data and whether a smart contract has been deployed. If there is enough data and a smart contract has been deployed, it can call it directly. Smart contract A completes the relevant transaction. If there is no smart contract, it sends the transaction to the transaction pool to wait for confirmation.

The transaction information stores the address of node B, the address of node A, the payment amount, and the timestamp. Miners select transactions from the transaction pool and package them into a block, which can be uploaded to the blockchain after other miners verify that the block is valid. Successful blockchain upload means the transaction is completed and node B sends its local encrypted data to node A through the transport layer.

Meanwhile, node B can also deploy smart contracts on the blockchain. Specifically, smart contracts are a big part of why blockchain is called "decentralized", allowing us to perform traceable, irreversible, and secure transactions without the need for third parties. Once a smart contract is linked, all nodes connected to the blockchain can execute this code locally, which performs obligations between the parties to the contract. In other words, node A can execute the smart contract deployed by node B on the blockchain to complete the transaction process of data, which makes the whole transaction more decentralized and intelligent.

In the above process, transactions between users do not require trust endorsement from a third party, and user data only needs to be stored locally, which greatly reduces the risk of data leakage. At the same time, users only need to upload their own address, hash value, and timestamp of the data, and then they can start transactions between other nodes, which greatly reduces the communication overhead of the IoT data flow in the blockchain network. It is worth noting that after the user uploads the data hash value, the hash value will be compared with the data hash value delivered during the transaction to verify whether the data has been tampered with, which further improves the reliability of the IoT data.

4. Implementation and Evaluation

4.1. Implementation Setup

The BDIDA-IoT refers to the source code of Bitcoin, uses Golang to implement a blockchain environment, and tests it on this platform. This section analyzes preliminary experimental data for BDIDA-IoT in a windows amd64 environment with an AMD Ryzen 5 5600H cpu with Radeon Graphics. We used a 35 KB data set containing 1000 blockchain addresses, which are base58-encoded 34-bit numbers generated by the Elliptic Curve Cryptography (ECC) algorithm. This means that our blockchain network can have up to 1000 nodes synchronized at the same time. In the performance test of blockchain, the number of blockchain nodes can greatly affect the consensus time, which in turn affects the overall system performance.

In the experiment to test the consensus time and network throughput of the blockchain, we set 25 to 40 nodes for testing, all nodes share 16 GB memory space, each transaction size in the blockchain network is 1 KB, and the data is transmitted using the TCP protocol. In the test DID generation and verification experiment, we test the time for 1 node to generate 1000 to 4000 DID Documents. According to the different standards of different DID schemes, the size of each DID Document in the BDIDA-IoT scheme is 700 B, the size of the DID Document in Baidu-DID is 1.3 KB, and the size of the DID Document in BSN-DID is 2 KB. In addition, the signature algorithm used is ECDSA.

It is worth mentioning that, owing to the inherent device heterogeneity within blockchain systems [24], experimental data often exhibit a degree of variability. When employing different consensus algorithms and varying the number of nodes in the blockchain network, significant disparities in the time taken for the overall blockchain to reach consensus arise. In our experiments, we opted for the PoW consensus algorithm for verification purposes. We adjusted the difficulty level to ensure that the time required to achieve consensus for an individual transaction remains within acceptable limits.

4.2. Evaluation

To implement our blockchain, we first set up the data structure in Listing 1. The Block structure is the same as the normal Block structure. Transaction structure represents the Transaction in our blockchain. It mainly includes Transaction type Form, Transaction ID,

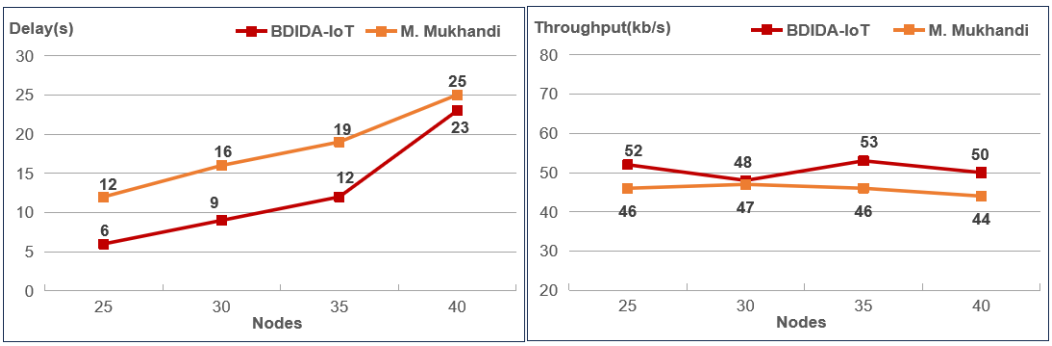
DataHash, and input and output. It is worth noting that the content of the transaction varies depending on the type of transaction. There are two types of payment transactions and stored transactions. If the transaction is a payment transaction, the DataHash defaults to 0, and Vin, Vout, and ID are set normally. When the transaction is a stored transaction, the Vin and Vout are empty because the stored transaction simply declares in the blockchain that it owns 100 MB of IoT data. DataHash is the hash value of 100 MB of data, which will be used to verify data integrity during the transaction. ID stores the user's identity information.

Listing 1. The main defination of Transaction struct.

```
type Transaction struct {
        ID        []byte
        Form      string
        DataHash  []byte
        Vin       []TxInput
        Vout      []TxOutput
}
```

4.2.1. Blockchain Latency and Throughput Analysis

Figure 8 shows the delay and throughput of this solution under different numbers of nodes. During the experiment, we observed that the delay increases almost linearly as the number of nodes increases, from 6 s for 25 nodes to 23 s for 40 nodes, and the delay results are better than the DDS-based scheme proposed by Mukhandi et al. [25]. For the same number of nodes, the overall performance improvement averaged 4.5 s. We took the PoW algorithm used in Bitcoin and adjusted the difficulty accordingly, while also reducing the redundancy of the block information, which helps reduce the time it takes the system to reach consensus. Because Mukhandi et al. 's scheme is tested on Ethereum, the block composition is more than our scheme with state tree, receipt tree, and other structures, which affects the efficiency of a single node when synchronizing blocks. At the same time, it can be seen from the throughput test of this scheme that the throughput result of our proposed system is better. There is not much change in the number of nodes. This depends on the data processing performance of the experimental equipment, the network environment where the equipment is located, and the communication protocol used. In our experiment, we had an average throughput of 50.75 kb/s for RPC communication using the TCP protocol.



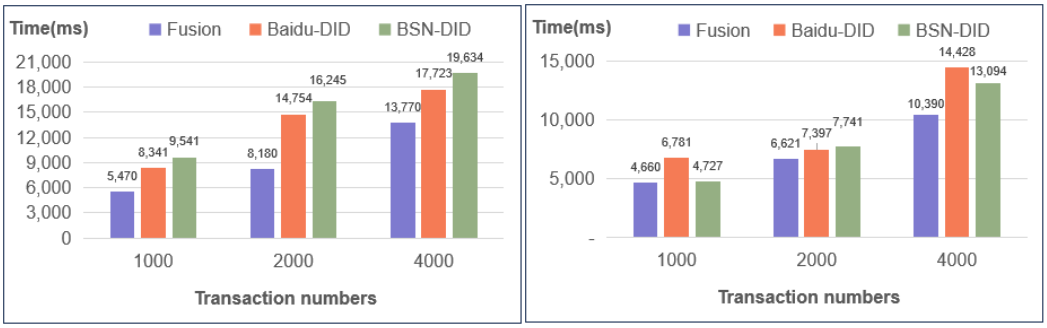
(**a**) (**b**)

Figure 8. The latency and throughput of transactions under the BDIDA-IoT and Mukhandi's [25] solution when the number of nodes is different. The abscissa is the number of nodes participating in the system, and the ordinate is the time it takes to reach consensus. (**a**) Delay Time shows the time it takes to reach consensus under different numbers of nodes. (**b**) Throughput shows the throughput data of the system under different numbers of nodes.

We have observed that the transaction completion delay rises proportionally with an increase in the number of nodes, and the extent of this increase becomes more pronounced. This phenomenon is attributed to variations in the volume of data [26]. Given that the throughput of each node remains relatively constant, it implies a limitation in the data processing capacity of individual nodes. As the number of nodes within the blockchain system grows, the data influx experienced by each node also escalates. Consequently, this surge in data has repercussions on the overall consensus achievement, necessitating a certain amount of time during the data transmission and reception processes.

4.2.2. DID Generation and Verification Time Analysis

Figure 9 shows the time taken to generate and validate a Verifiable Credential (VC) for different scenarios. It can be seen that, in our scheme, the time to generate a VC and the time to verify a VC are lower than those of BSN and BaiDu. The VC needs to be signed by the owner of the DID. Different VCs needs different information to be verified. When there is more information to be verified, the overall generation and verification process will take more time. In our scenario, the contents of the VC encompass the message, its corresponding message digest, a signature confirming the endorsement by the controller, details of the VC issuer, and the verification URL designated by the issuer specifically for this VC. At the same time, BIDU-DID and BSN-DID are relatively static, which means that whenever the data content or information needs to be changed a new DID needs to be applied for. However, the content of DID in our scheme is modifiable, which means that a single generation is reused many times, and this mechanism helps to improve the performance of DIDs. In our solution, both the generation of DIDs and the verification of a VC fall within an acceptable time-frame. At the same time, in different scenarios the generation of a DID will vary from the information contained in the verification. In some single-purpose verification scenarios, the overall time consumption can be further reduced.



(**a**) (**b**)

Figure 9. Comparison of the VC generation and verification times of the BDIDA-IoT solution with the Baidu [27,28] and BSN [29,30] DID solutions. The abscissa represents the number of VCs and the ordinate represents the total time (ms) taken to complete generation or verification. (**a**) VC Generation Time shows the experimental results of VC generation time. (**b**) VC Verification Time shows the experimental results of VC verification time.

Every transaction within our blockchain is recorded, while the transmission of IoT data takes place through alternative channels. Through the blockchain, users possess the capability to inquire about the data holdings of others and initiate transactions with the respective data owners. Following the successful mining of transaction information onto the blockchain, data owners have the flexibility to choose an optimal method for data delivery.

This entire transaction process, as delineated above, can be seamlessly executed by deploying smart contracts on the blockchain infrastructure. Algorithm 1 encapsulates part of the logic of a smart contract designed for transaction execution. Leveraging smart contracts significantly enhances the efficiency of transactions. In addition, automatic verification of

data integrity injects a layer of trust and automation throughout the transaction process, which can simplify the process of trading.

Algorithm 1 Part of the logic of smart contracts

INPUT: VC (Verifiable Credential)
RESULT: Balance (Sender and Receiver)
$Sender\_balance \leftarrow Sender\_balance - count$
$Receiver\_balance \leftarrow Receiver\_balance + count$
if $VC.Signature.Verify() =$ true **then**
    $count \leftarrow VC.count$
    $VC.State \leftarrow$ done
    AddBalance()
else
    return error.False("no permission")
end if

4.2.3. Analysis of Computational Overhead and Memory Usage

During the experiment, the experimental data may be affected by CPU performance, network bandwidth, number of nodes, and data IO speed. The computing power of the CPU will affect the execution time of the consensus algorithm of the node and also affect the time of the node to generate and verify the DID. Network bandwidth affects the speed at which nodes send and receive transactions to the blockchain network, which increases block processing time and reduces system performance. The number of nodes increases the amount of data in the network, and as the number of nodes increases so does the time it takes to synchronize blockchain data between nodes. The data I/O speed affects the time the node takes to process the data in the transaction pool, thereby reducing the block time.

After the framework is deployed to IoT devices, the central server responsible for managing resource-constrained IoT devices has resource constraints and performance constraints. Resource-constrained IoT devices are responsible for collecting data and delivering it to the central server, which participates in the blockchain network as a blockchain node.

In terms of computational overhead and energy consumption, since we incorporate decentralized identity to ensure trust issues among users in the blockchain network, the difficulty of reaching consensus on each block can be reduced. This greatly reduces the computing overhead and energy consumption of nodes, and also shortens the time for reaching consensus between nodes.

In terms of memory usage, nodes need to maintain a transaction pool and DSS locally. The transaction pool includes transaction requests sent to the blockchain network, and the DSS stores the transaction flow process of the data type. The maximum block size is 1 MB, including a fixed block header of 80 B, and the transaction body is responsible for storing transactions. The size of transactions is usually around 250 B, which means that a block can store up to 4000 transactions. Therefore, a node only needs to have the ability to store more than 4000 transactions to perform block construction work normally and participate in the blockchain network normally. In the early stages of blockchain network operation, memory consumption is extremely low for the burden on nodes.

5. Conclusions

In this paper, we propose a blockchain-based trusted data management and trading platform for the IoT. After analyzing the credibility of the platform, the data provider, and the data provider's autonomy to the data in the traditional data management mode of the IoT, we consider two kinds of transactions, namely payment transactions and storage transactions, and realize the up-chain and trusted transaction of the IoT data by utilizing the characteristics of blockchain. At the same time, in order to improve the transaction automation and efficiency, we designed a smart contract to complete this process. Finally, we implemented the proposed system.

While the results in the experimental test show positive results, in the actual IoT data flow scenario the participating transactions do not belong to the same network environment, so the time to reach consensus on the blockchain will increase. At the same time, our tests of the system were evaluated with a small number of nodes. In a real-world scenario, the number of nodes participating in the blockchain would be higher, which would increase the data processing burden on each node and reduce performance. In the process of running the blockchain system, it may face the problem of increased demand, so it is also necessary to consider the problem of system scalability and update the version in a more convenient way.

In terms of system scalability, we have layered the system, and scalability can be deployed at the application layer in the form of plug-ins. At the same time, our blockchain platform is modified with reference to the source code of Bitcoin, and relevant data structures can be flexibly modified to adapt to more needs. In the future, we will make various functions of the blockchain into APIs, and the application layer only needs to use these APIs to achieve good scalability.

Future works can be improved in the following ways:

(1) The experiment in this article was conducted on a single machine and multiple nodes. In the future, we will consider using clusters or edge computing services for deployment to further verify the performance of the system.
(2) In the future, more physical equipment can be used to test the reliability and throughput of the system.
(3) Future research can try to improve scalability and support more IoT application integration. At the same time, the content of our blockchain platform is modifiable, which means we can expand more services at the application layer through plug-ins.
(4) Future research can consider adding a federated learning framework to deal with the distributed training scenarios of large-scale IoT data and achieve this expansion by dividing the roles of nodes in the blockchain system.

Author Contributions: Conceptualization, Z.Y. and Y.L.; methodology, Z.Y., Y.L. and X.J.; validation, Z.Y. and Y.L.; writing—original draft preparation, Z.Y.; writing—review and editing, X.J., X.L., Y.X. and M.L.; supervision, Y.L., X.J. and X.L.; project administration, Y.L., X.J., X.L., P.C., B.T. and B.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 62302317 and 62302316), Shenzhen Colleges and Universities Stable Support Program (Grant Nos. 20220715183602001 and 20231122005530001), Shenzhen Science and Technology Program (Grant No. RCBS20221008093227027), Natural Science Foundation of Top Talent of SZTU (Grant No. GDRC202319), and SZTU Project No. 20224027010006.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors. The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Abdelmaboud, A.; Ahmed, A.I.A.; Abaker, M.; Eisa, T.A.E.; Albasheer, H.; Ghorashi, S.A.; Karim, F.K. Blockchain for IoT applications: Taxonomy, platforms, recent advances, challenges and future research directions. *Electronics* **2022**, *11*, 630. [CrossRef]
2. Yadav, K.; Kariri, E.; Alotaibi, S.D.; Viriyasitavat, W.; Dhiman, G.; Kaur, A. Privacy protection against attack scenario of federated learning using internet of things. *Enterp. Inf. Syst.* **2023**, *17*, 2101025. [CrossRef]
3. Yuan, B.; Ge, S.; Xing, W. A federated learning framework for healthcare IoT devices. *arXiv* **2020**, arXiv:2005.05083.
4. Berdik, D.; Otoum, S.; Schmidt, N.; Porter, D.; Jararweh, Y. A survey on blockchain for information systems management and security. *Inf. Process. Manag.* **2021**, *58*, 102397. [CrossRef]
5. Balcerzak, A.P.; Nica, E.; Rogalska, E.; Poliak, M.; Klieštik, T.; Sabie, O.M. Blockchain technology and smart contracts in decentralized governance systems. *Adm. Sci.* **2022**, *12*, 96. [CrossRef]

6. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Jolfaei, A.; Islam, A.N. A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system. *J. Parallel Distrib. Comput.* **2023**, *172*, 69–83. [CrossRef]
7. Banerjee, M.; Lee, J.; Choo, K.K.R. A blockchain future for internet of things security: A position paper. *Digit. Commun. Netw.* **2018**, *4*, 149–160. [CrossRef]
8. Azbeg, K.; Ouchetto, O.; Andaloussi, S.J. BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. *Egypt. Inform. J.* **2022**, *23*, 329–343. [CrossRef]
9. Kim, B.G.; Cho, Y.S.; Kim, S.H.; Kim, H.; Woo, S.S. A security analysis of blockchain-based did services. *IEEE Access* **2021**, *9*, 22894–22913. [CrossRef]
10. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, B.; et al. Towards federated learning at scale: System design. *Proc. Mach. Learn. Syst.* **2019**, *1*, 374–388.
11. Maftei, A.A.; Lavric, A.; Petrariu, A.I.; Popa, V. Massive data storage solution for IoT devices using blockchain technologies. *Sensors* **2023**, *23*, 1570. [CrossRef]
12. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin* **2008**, *4*, 15.
13. Kunduru, A.R. Blockchain Technology for ERP Systems: A Review. *Am. J. Eng. Mech. Archit.* **2023**, *1*, 56–63.
14. Bao, Q.; Li, B.; Hu, T.; Sun, X. A survey of blockchain consensus safety and security: State-of-the-art, challenges, and future work. *J. Syst. Softw.* **2023**, *196*, 111555. [CrossRef]
15. Reed, D.; Sporny, M.; Longley, D.; Allen, C.; Grant, R.; Sabadello, M.; Holt, J. *Decentralized Identifiers (Dids) v1.0;* Draft Community Group Report; World Wide Web Consortium, Inc.: Wakefield, MA, USA, 2020.
16. Fotiou, N.; Thomas, Y.; Siris, V.A.; Xylomenos, G.; Polyzos, G.C. Self-verifiable content using decentralized identifiers. *Comput. Netw.* **2023**, *230*, 109799. [CrossRef]
17. Dixit, A.; Smith-Creasey, M.; Rajarajan, M. A Decentralized IIoT Identity Framework based on Self-Sovereign Identity using Blockchain. In Proceedings of the IEEE 47th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 26–29 September 2022; pp. 335–338.
18. Fotiou, N.; Thomas, Y.; Xylomenos, G.; Siris, V.A.; Polyzos, G.C. Authentication and Authorization for Content-Centric Routing using W3C DIDs and VCs. In Proceedings of the IEEE Conference on Standards for Communications and Networking (CSCN), Thessaloniki, Greece, 28–30 November 2022; pp. 163–168.
19. Fan, X.; Chai, Q.; Xu, L.; Guo, D. DIAM-IoT: A decentralized identity and access management framework for internet of things. In Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure, Taipei, Taiwan, 6 October 2020; pp. 186–191.
20. Zhu, X.; He, D.; Bao, Z.; Luo, M.; Peng, C. An efficient decentralized identity management system based on range proof for social networks. *IEEE Open J. Comput. Soc.* **2023**, *4*, 84–96. [CrossRef]
21. Dib, O.; Toumi, K. Decentralized identity systems: Architecture, challenges, solutions and future directions. *Ann. Emerg. Technol. Comput. AETiC* **2020**, *4*, 19–40. [CrossRef]
22. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
23. Steichen, M.; Fiz, B.; Norvill, R.; Shbair, W.; State, R. Blockchain-based, decentralized access control for IPFS. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1499–1506.
24. Tseng, L.; Wong, L.; Otoum, S.; Aloqaily, M.; Othman, J.B. Blockchain for managing heterogeneous internet of things: A perspective architecture. *IEEE Netw.* **2020**, *34*, 16–23. [CrossRef]
25. Mukh, i M.; Damião, F.; Granjal, J.; Vilela, J.P. Blockchain-based device identity management with consensus authentication for IoT devices. In Proceedings of the IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 433–436.
26. Khashan, O.A.; Khafajah, N.M. Efficient hybrid centralized and blockchain-based authentication architecture for heterogeneous IoT systems. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 726–739. [CrossRef]
27. Baidu-DID. Available online: https://did.baidu.com (accessed on 4 January 2024).
28. Jing, Y.; You, X.; Bi, D.; Li, H. The decentralized identity and its application for industrial internet. In Proceedings of the 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 10–12 December 2021; pp. 671–674.
29. BSN-Spartan-DID. Available online: https://github.com/BSN-Spartan/DID (accessed on 4 January 2024).
30. Hendershott, T.; Zhang, X.; Zhao, J.L.; Zheng, Z. FinTech as a game changer: Overview of research frontiers. *Inf. Syst. Res.* **2021**, *32*, 1–17. [CrossRef]