applied sciences

MDPI

Article

Research on Secure Storage Technology of Spatiotemporal Big Data Based on Blockchain

Bao Zhou [1,2,3,4], Junsan Zhao [1,2,3,4,*], Guoping Chen [1,2,3,4] and Ying Yin [5]

[1] Faculty of Land Resources Engineering, Kunming University of Science and Technology, Kunming 650093, China; zhoubao@stu.kust.edu.cn (B.Z.); chenguoping@kust.edu.cn (G.C.)
[2] Key Laboratory of Geospatial Information Integration Innovation for Smart Mines, Kunming 650093, China
[3] Spatial Information Integration Technology of Natural Resources in Universities of Yunnan Province, Kunming 650211, China
[4] The Industry-University-Research Integration Innovation Base of Natural Resources Smart Management, Kunming 650211, China
[5] College of Electronic and Information Engineering, West Anhui University, Lu'an 237000, China; 03000122@wxc.edu.cn
* Correspondence: 11301057@kust.edu.cn

Abstract: With the popularity of spatiotemporal big data applications, more and more sensitive data are generated by users, and the sharing and secure storage of spatiotemporal big data are faced with many challenges. In response to these challenges, the present paper puts forward a new technology called CSSoB (Classified Secure Storage Technology over Blockchain) that leverages blockchain technology to enable classified secure storage of spatiotemporal big data. This paper introduces a twofold approach to tackle challenges associated with spatiotemporal big data. First, the paper proposes a strategy to fragment and distribute space–time big data while enabling both encryption and nonencryption operations based on different data types. The sharing of sensitive data is enabled via smart contract technology. Second, CSSoB's single-node storage performance was assessed under local and local area network (LAN) conditions, and results indicate that the read performance of CSSoB surpasses its write performance. In addition, read and write performance were observed to increase significantly as the file size increased. Finally, the transactions per second (TPS) of CSSoB and the Hadoop Distributed File System (HDFS) were compared under varying thread numbers. In particular, when the thread number was set to 100, CSSoB demonstrated a TPS improvement of 7.8% in comparison with HDFS. Given the remarkable performance of CSSoB, its adoption can not only enhance storage performance, but also improve storage security to a great extent. Moreover, the fragmentation processing technology employed in this study enables secure storage and rapid data querying while greatly improving spatiotemporal data processing capabilities.

Keywords: blockchain; spatiotemporal big data; transactions per second; Hadoop Distributed File System

Citation: Zhou, B.; Zhao, J.; Chen, G.; Yin, Y. Research on Secure Storage Technology of Spatiotemporal Big Data Based on Blockchain. *Appl. Sci.* **2023**, *13*, 7911. https://doi.org/10.3390/app13137911

Academic Editor: Gianluca Lax

Received: 9 June 2023
Revised: 2 July 2023
Accepted: 3 July 2023
Published: 6 July 2023

1. Introduction

In the current era of big data, the exponential increase in data volume has spurred researchers to explore the application of big data that include sensitive and private information [1–3]. The integrity of data is crucial to ensure that accurate knowledge can be extracted from secure big data analysis. Regular collection and analysis of spatiotemporal data are carried out in the big data environment. However, the escalating security risks (e.g., big data privacy, quality, and security mechanisms) pose a significant threat to these applications [4].

In the storage of big data, spatiotemporal big data are commonly stored in a distributed file system. In a distributed storage system, multiple nodes must collaborate to perform specific tasks. As such, compromising one or more nodes can undermine the accuracy of

computation results, and management becomes more complex. Consequently, traditional asymmetric and symmetric encryption techniques are challenging to directly apply in big data storage solutions [5]. In light of this, existing big data platform audits have become inadequate, leaving them vulnerable to numerous security challenges.

Ensuring spatiotemporal big data validity and signatures is crucial to secure the storage of big data [6]. Data integrity becomes a primary security concern for users since they no longer have control over their data once they upload them. Cloud storage enables applications to effectively manage their remote data but is also vulnerable to tampering. In cloud storage, directly encrypting large data sets increases the risk of key management and requires significant computing overhead due to the large scale, large increment, and fast change of data sets [7]. If the secret key is exposed, the entire data set can be compromised and stolen. Blockchain technology has drawn attention from academia and industry due to its characteristics, such as immutability, confidentiality, integrity, and authorization [8].

In reference to the literature [9], a public audit scheme was developed for verifying the integrity of cloud storage data using blockchain technology. Data owners store lightweight verification tags on the blockchain and generate proofs by building a Merkle hash tree using these tags to minimize computation and communication overhead during integrity verification. The literature [10] proposes a blockchain-based multirobot collaborative application framework and potential benefits to combat the COVID-19 epidemic, such as monitoring and outdoor and hospital end-to-end delivery systems. The challenges and opportunities of integrating blockchain, multirobot collaboration, and the Internet of Things to address COVID-19 and future pandemics were also discussed. The researchers propose a conceptual framework to address digital twin collaboration needs through the use of blockchain, predictive analytics, and digital twin technology [11], a ledger-based digital twin framework that focuses on real-time operational data analysis and distributed consensus algorithms. It also describes how the conceptual framework can be applied using use cases of intelligent transportation systems, namely, intelligent logistics and railway predictive maintenance.

A novel solution for big data security was proposed by Alhazmi et al. in 2022 [12], which utilized blockchain technology and fragmentation to construct a security framework. In this framework, data fragments are distributed and stored within the big data environment to add an additional layer of data protection. Ren et al. presented the blockchain-based secure storage mechanism, which is designed to address the issue of limited blockchain storage capacity. The model employs an on-chain and off-chain collaborative storage approach, where on-chain and off-chain security authentication protocols are constructed using updatable subvector commitments, ensuring the consistency of on-chain and off-chain data, and enabling batch processing capabilities [13].

The ability of data to perform batch processing notwithstanding, the storage and security of spatiotemporal big data present significant challenges [14–16]. Data owners of-ten outsource crucial spatiotemporal data to cloud servers for storage, which leads to a loss of physical control over the data. As a result, the cloud storage server can tamper with the stored spatiotemporal data at will.

Therefore, the space–time of big data outsourcing is always facing great risks. To ad-dress these challenges, blockchain technology has emerged as an important solution due to its ability to solve modern technical issues, such as decentralization, immutability, trust, data ownership, and traceability [17]. However, the sheer size and diversity of big spatiotemporal data mean that they can quickly consume the storage capacity of blockchain nodes, leading to system performance degradation. Thus, it is challenging to directly store vast amounts of spatiotemporal data on the blockchain.

Considering the aforementioned challenges, the need for a secure and dependable storage solution for large-scale spatiotemporal data has become increasingly pressing. To address this issue and ensure real-time data security and reliability, a blockchain-based solution is proposed in this study. The solution entails an on-chain and off-chain collaborative mechanism for the secure storage of big spatiotemporal data. Specifically, the sensitive

data of spatiotemporal big data are stored on the blockchain using a specific content ID (CID), while nonsensitive data are directly stored in the database. The blockchain verifier exhibits the capacity to undertake batch queries, batch verification, and batch updates for large spatiotemporal data. The mechanism also enables the recording of data update values by the spatiotemporal data owner to the achieve traceability of big spatiotemporal data.

This article also encrypts sensitive data to be uploaded to IPFS. First, the sensitive data are encrypted by an elliptic curve. Second, the encrypted data are divided into multiple data blocks, each of which has a corresponding CID. Then the CID of the data block is randomly arranged, and the randomly arranged CID is saved in the metadata. Finally, the metadata are encrypted by an elliptic curve. The encrypted data and metadata are uploaded to IPFS and blockchain, respectively. In this way, the content of sensitive data is encrypted. Additionally, the TPS performances of the InterPlanetary File System (IPFS) and HDFS are compared, and the test results indicate a significant improvement in storage security when using the IPFS approach.

2. Encryption Principle

2.1. Elliptic Curve Cryptography

In the past few years, blockchain technology has gained significant attention due to its association with digital currencies. The unique features of blockchain technology, such as decentralization, nontampering, nonforgeability, and traceability, have been thoroughly researched [18]. By applying blockchain technology to the secure storage of spatiotemporal big data, the limitations of traditional centralized models can be overcome, and the security of spatiotemporal big data can be ensured. The Internet of Things and the Internet have led to the generation of an enormous amount of spatiotemporal big data worldwide. Therefore, it has become an urgent task to ensure the storage security of spatiotemporal big data.

Elliptic curve cryptography (ECC) was first discovered in 1985 by Neil et al. The security of the ECC cryptosystem is based on the intractability of the elliptic curve discrete logarithm problem. ECC has several advantages compared with other public key cryptosystems, such as RSA, DSA, and DH. ECC can use smaller elliptic curve groups while maintaining the same security level, resulting in smaller key sizes, reduced storage and transmission requirements, and faster operations. These advantages of ECC not only solve implementation difficulties that arise from increasing the key length of other public key cryptosystems to enhance the security of the cryptosystem, but also enable fast encryption and decryption, digital signatures, key exchange, identity authentication, and other applications [19]. The specific applications of ECC can be seen in the following aspects:

(1) Applications within ECDSA: The elliptic digital signature algorithm first produces a message digest through a hash function, followed by signature computation using the elliptic curve. Private keys are utilized for signing, while public keys serve verification purposes [20].
(2) Deployment in wireless networks: Certificates encompass the public key and signature of both the certificate issuer and the certificate holder, with the signature relying on ECC [21].
(3) Implementation on web servers: This approach curtails the data and time transmitted between the system and application program, thus enhancing practicality; it conserves time and bandwidth, while addressing compatibility issues more effectively through algorithm negotiation [22].
(4) Utilization in IC cards: In the absence of a dedicated processor, ECC swiftly and securely executes digital signatures on IC cards, a feat difficult to achieve in RSA systems. Additionally, ECC minimizes the program code, key length, and certificate size, substantially reducing IC card costs [23].

Addressing the issue of secure storage for blockchain spatiotemporal big data, this paper introduces a secure storage scheme founded on elliptic curve encryption.

2.2. The Encryption Theory of Elliptic Curve

Weierstrass equation [24,25]:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \tag{1}$$

The constructed plane curve is called an elliptic curve, denoted as E, where a, b, c, d, and e, $\in Fp$; Fp is a finite field. The points on the elliptic curve are all on the Fp field, and a point O at infinity is also defined. The point O defined in the field at infinity corresponds to the coordinates (0, 0).

For the finite field Fp, when its eigenvalue is greater than 3, the Weierstrass equation can be transformed into

$$y^2 = x^3 + ax + b \tag{2}$$

Then the elliptic curve on the finite field Fp can be defined as $Ep(a,b)$ by a and b, p is a prime number, $x,y \in [0, p-1]$, and $4a^3 + 27b^2 \neq 0$.

An additive group operation, "+", is defined on E, with its geometric meaning represented as follows: given two points, P and Q, on the elliptic curve, connect P and Q with a straight line, l; if $P = Q$, then l is the tangent at point P on the elliptic curve. R is the symmetric point where the straight line l intersects the curve at another point R', after passing through P and Q on the x-axis; R' is the inverse element. Then, $(E, +)$ constitutes an abelian group, with O as the additive identity element, as depicted in Figures 1 and 2.

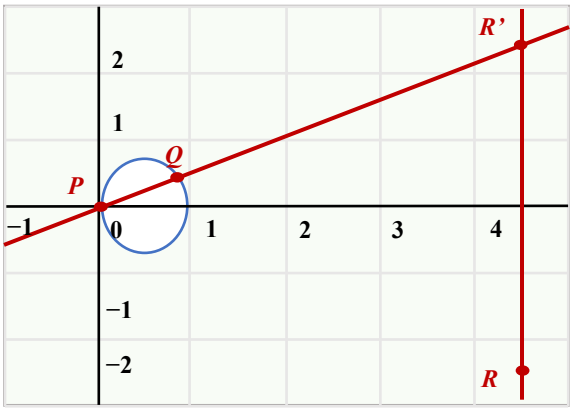

Figure 1. Geometric addition.

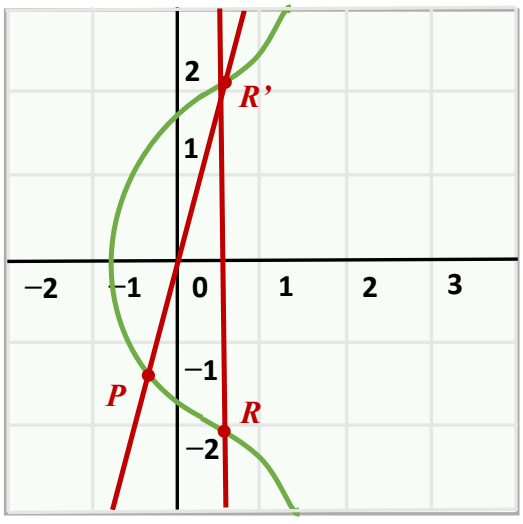

Figure 2. Geometric doubling.

For finite fields Fp, the addition operation on elliptic curves is defined as follows. Suppose a point $P = (x1, y1)$ and $Q = (x2, y2)$ on E.

Null:

$$O + P = P,\ O + O = O\ (\text{point at infinity}) \tag{3}$$

Click to add:

$$P + Q = \begin{cases} O & x_1 = x_2\ and\ y_1 = -y_2 \\ (x_3,\ y_3) & others \end{cases} \tag{4}$$

$$in, k = \begin{cases} (3x_1 + a)/2y_1 (mod\ p) & P = Q \\ (y_2 - y_1)/(x_2 - x_1)\ (mod\ p) & P \neq Q \end{cases} \tag{5}$$

$$x_3 = k^2 - x_1 - x_2\ (mod\ p) \tag{6}$$

$$y_3 = k(x_1 - x_3) - y_1\ \ (mod\ p) \tag{7}$$

Doubling:

$$if\ P = Q, then\ P + Q = 2P \tag{8}$$

Dot multiplication:

$$kP = P + P + \cdots \ldots + P \tag{9}$$

2.3. Principles of Elliptic Curve Cryptography

As depicted in Figure 3, regarding the elliptic curve encryption process, a finite field Fp is selected, and the elliptic curve $Ep(a,b)$ and a base point P on it, with order being a prime number n, are determined. Ellipse parameters a and b, n, and P are public information.

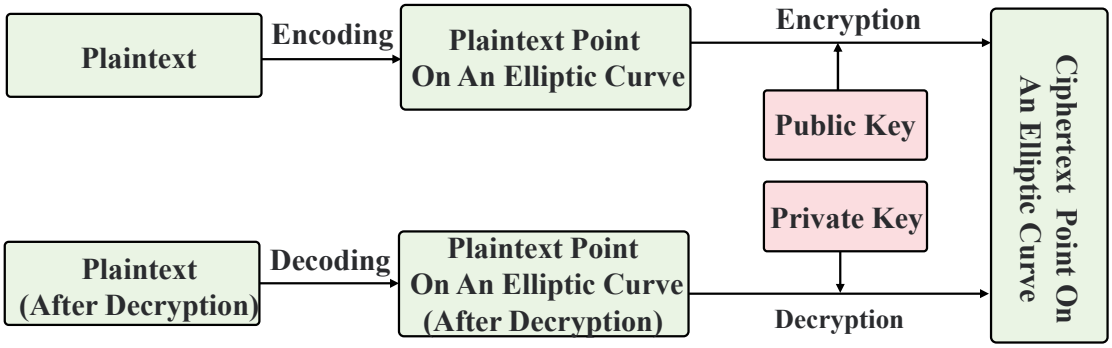

Figure 3. Elliptic curve encryption and decryption process.

Step 1: Key generation

Select the base point P on the elliptic curve $Ep(a,b)$, choose a large number k as the private key, and generate the public key $Q = kP$.

Step 2: Encryption

When B sends the plaintext to A, the following steps are performed:

- Pass $Ep(a,b)$ and points Q and P to A;
- Encode the plaintext to be transmitted to a point M on $Ep(a,b)$;
- Generate a random integer r ($r < n$);
- Calculation $C = \{rP, M + rQ\}$;

Step 3: Decrypt

- $M + rQ - k(rP) = M + r(kP) - k(rP) = M$;
- Decode the point M to obtain the plaintext.

2.4. Performance Analysis

In order to verify the performance of elliptic curve encryption, this paper uses MATLAB for simulation verification, mainly considering the influence of the plaintext length and key k value on the cost of encryption time, as shown in Figures 4 and 5.

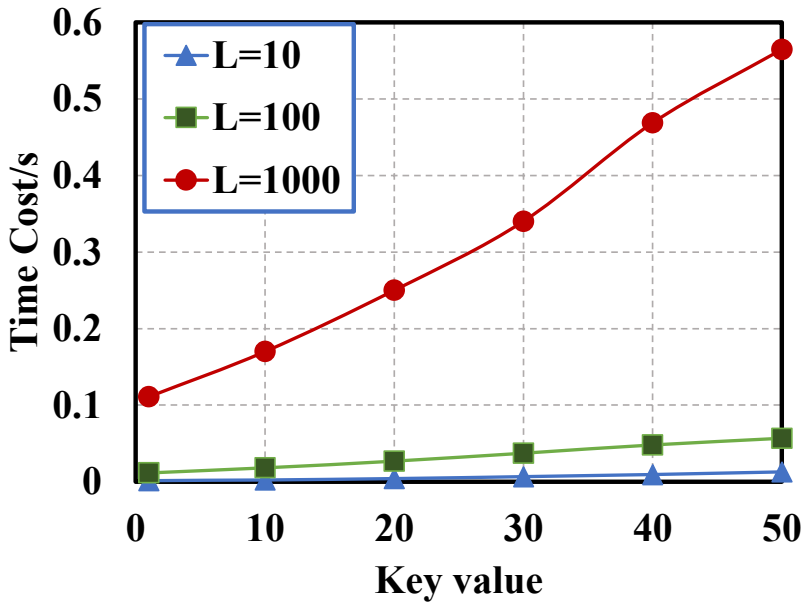

Figure 4. Relationship between time cost and key k value change.

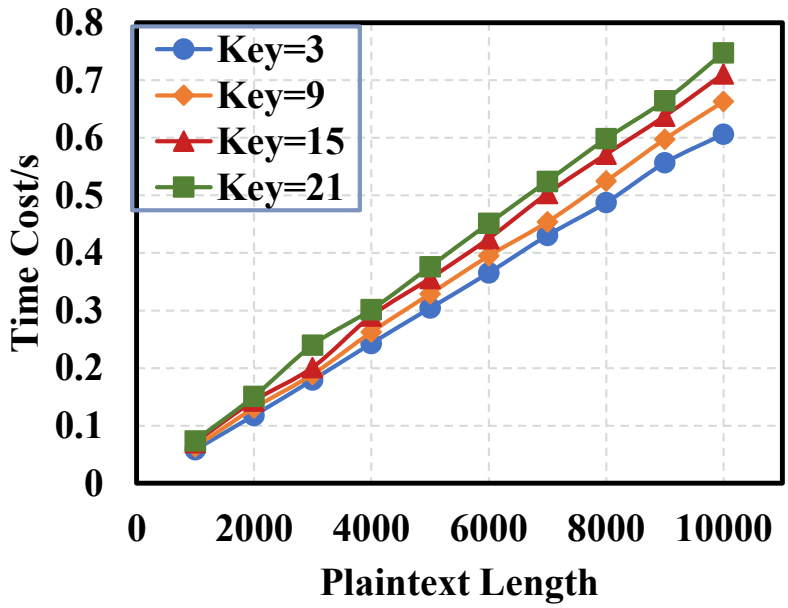

Figure 5. Relationship between plaintext and time cost.

Considering the influence of the plaintext length on encryption performance, a plaintext with a fixed length of L = 10, 100, 1000 is selected for elliptic curve encryption and decryption. L indicates the number of characters from 0 to 9. A total of 10 characters are in a group. For example, if L = 100, it is 10 groups of digits from 0 to 9. As shown in Figure 4, the time spent for elliptic curve encryption and decryption increases with the increase in equal intervals of key values. Therefore, for a plaintext with a large amount of data, a smaller k value should be selected to save time.

On the basis of Figure 4, considering the influence of the plaintext length and key k value on encryption performance, 10 plaintexts of different lengths and 4 different k values

are selected for encryption and decryption, as shown in Figure 5. With the same k value, the time cost of elliptic curve encryption and decryption is positively correlated with the plaintext length as the plaintext length increases. Similarly, when the plaintext length is the same, the larger the key k value, the more time it takes. The difference between the four curves with different k values is not large, which further indicates that the k value has little influence on the time cost of encryption and decryption, while the plaintext length has great influence on it.

The results show that when the elliptic curve is used for encryption, it is necessary to consider the influence of the plaintext length and k value, and select the appropriate value to obtain better encryption performance and save time.

3. Introduction to Related Technologies of the Scheme

3.1. Spatiotemporal Big Data Technology

Spatiotemporal big data represent a voluminous data set encompassing both temporal and spatial dimensions, which has found broad applications across multiple disciplines. The data can be categorized as government and corporate data, search data, cyberspace data, spatial media data related to location, and human geography data, as illustrated in Figure 6. This type of big data is primarily composed of information relevant to national security, industry, meteorology, transportation, medical care, social networking, among other areas. To analyze and process spatiotemporal big data, it is essential to investigate reliable, effective, and practical techniques for data management and processing. For specific spatiotemporal data, such as those involving national defense, medical care, and industry, personal privacy and the vitality of national energy resources must be safeguarded. Therefore, the integration, cleaning, transformation, and extraction of spatiotemporal big data are crucial preprocessing steps requiring effective approaches.

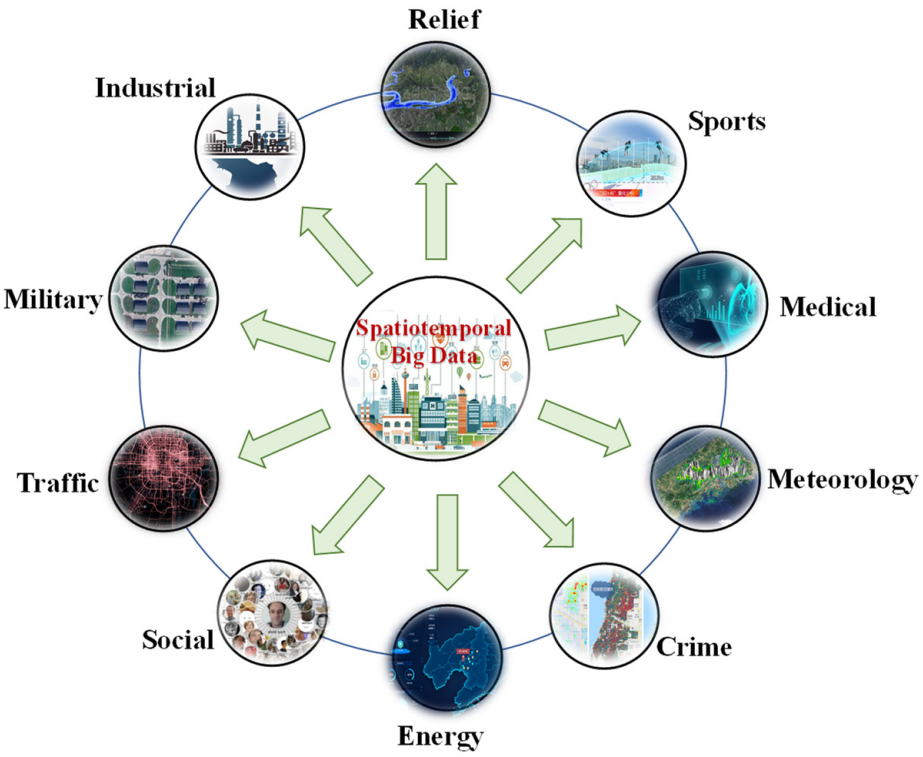

Figure 6. Classification of spatiotemporal big data.

The secure storage of spatiotemporal big data is also a significant challenge. To ensure data security, technologies such as encryption, backup, recovery, access control, and authentication can be employed. Moreover, blockchain technology is a promising solution for secure data storage, offering a tamper-proof distributed database that guarantees

the security and integrity of spatiotemporal big data. To summarize, the processing of spatiotemporal big data necessitates a comprehensive consideration of data management, processing, and secure storage, requiring corresponding technical means and measures to ensure the data's integrity and security. Additionally, the utilization of blockchain technology can be explored to address the security storage issue.

3.2. IPFS Storage Technology

IPFS is a protocol with the following functions:

(1) Content addressing: Contrasting with the address addressing in the HTTP protocol, IPFS employs multiple hashes to uniquely identify a data block's content. Consequently, hash-based addressing is employed during searches.
(2) Tamper resistance: The unique hash facilitates determining whether the data block's content has been tampered with.
(3) Deduplication: Relying on the premise that identical data share the same hash, duplicate data are identified to conserve storage space.
(4) Content distribution: Content distribution is achieved through BitTorrent, a content distribution protocol. BitTorrent utilizes content distribution and peer-to-peer technology to enhance the efficiency of large file sharing among users, reducing the burden on centralized servers.
(5) P2P transmission: Transmissions primarily depend on libp2p, which is responsible for IPFS data's network communication, routing, exchange, and other functions.

Compared with the traditional HTTP protocol, IPFS exhibits the following advantages:

(1) IPFS eliminates reliance on backbone networks and centralized servers, connecting all devices in the network through a file system. This enables swift access to files stored on the system from any location worldwide.
(2) IPFS features historical version tracking, facilitating permanent data storage and ensuring all operations are reversible, akin to the widely used git.
(3) IPFS can transform the distribution mechanism of web content, decentralizing it and employing content delivery network (CDN) acceleration technology.
(4) Within the IPFS system, files and data are unique. When a file joins the IPFS network, a unique encrypted hash value is assigned to the content based on calculations.
(5) IPFS possesses a file duplication detection mechanism, thereby preventing resource redundancy issues.

4. The Principle of Safe Storage Technology of Spatiotemporal Big Data Blockchain

4.1. Introduction to Secure Storage Framework

This paper presents a novel approach to decentralized authentication and data auditing of sensor nodes, utilizing the IPFS system and blockchain technology. The content-addressable feature of IPFS is leveraged to address the storage constraints of blockchain. IPFS stores the encrypted data (or files) in the peer-to-peer nodes of the decentralized network, and returns the storage address of the encrypted data straddr. This straddr is used to retrieve encrypted data in IPFS. After the successful validation of ON, a block is created by SN and undergoes the block validation process. ON indicates an ordinary sensor node, and SN indicates a sink node. In our network model, a blockchain network is maintained by all deployed SNs, as shown in Figure 7, which depicts the framework of the blockchain secure storage structure. The figure provides an explanation of the numerical symbols used.

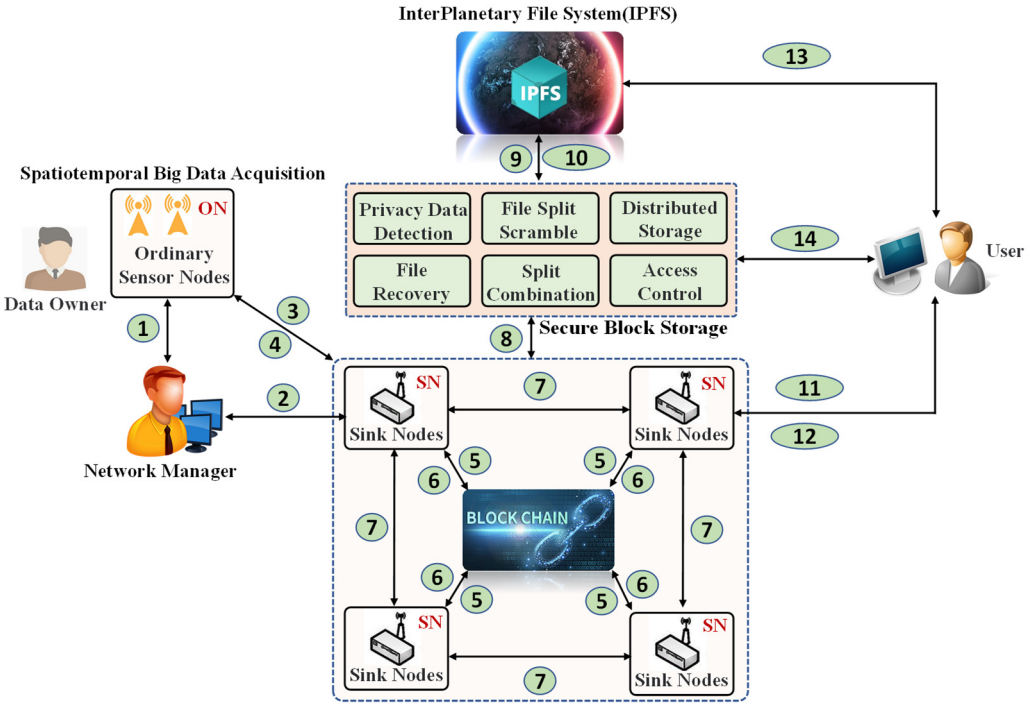

Figure 7. Blockchain secure storage framework.

Data Owner provides raw data through ON. ON collects original data, and SN collects data of sensor nodes. The security block storage module in SN encrypts Data in blocks to obtain the original Id Data and encrypted data. Id Data are stored in the blockchain, and encrypted data are stored in IPFS. When a user requests Data, the sink node requests the blockchain and IPFS to obtain the Id Data and the encrypted data, which are then decrypted back into the raw data.

Step 1: Represents the registration node for common sensors, registering with the network administrator.

Step 2: Signifies the sink node, composed of ordinary sensor nodes.

Step 3: Denotes the spatiotemporal data information gathered by the sensor, generating ordinary nodes and sink nodes.

Step 4: Involves the aggregation node performing identity authentication on common nodes.

Step 5: Depicts the verified block being uploaded to the blockchain network, recording the information.

Step 6: Demonstrates the blockchain network returning a specific Id-block to the sink node.

Step 7: Encompasses mutual authentication between sink nodes and verification of new blocks for sink nodes.

Step 8: Represents Data Owner (DO) uploading data and Id-block, while Secure Block Storage (SBC) gathers segmented data segments and encrypts private data.

Step 9: Illustrates SBC uploading encrypted and partitioned data information to IPFS for reconstruction and decryption.

Step 10: Indicates IPFS returning the hash value of encrypted data to the respective sink nodes for reconstruction and decryption.

Step 11: Involves users requesting hash values from their corresponding sink nodes.

Step 12: Denotes the sink node providing the hash value to the data requester.

Step 13: Signifies the data requester requiring the original data corresponding to the hash value from IPFS.

Step 14: Demonstrates IPFS providing data to SBC and reconstructing decrypted data. Ultimately, SBC sends the data to the end user.

4.2. Data Upload and Download

The secure storage process for spatiotemporal big data is depicted in Figure 8, which comprises two primary processes, namely, the upload and download processes. Figure 8a shows the upload of spatiotemporal big data. To begin with, the spatiotemporal big data are first encoded, as exemplified by #000, #001, #010, #011, #100, #101, #101, and #111, which represent diverse spatiotemporal data file information. Data security storage is approached by taking into account the identification of data stealth. In the case where the spatiotemporal big data entail private data, such as military, industrial, medical, and other information, they are first encrypted and subsequently stored in blocks. The data undergo direct scrambling into blocks, with the scrambled data being encoded as follows: #001, #011, #010, #000, #110, #101, #111, and #100. Block information Id-Data are then added, and the block information is stored in the blockchain, while the corresponding data are stored in IPFS. This methodology reduces the data cost and running time of the blockchain, while simultaneously recording block information in the blockchain network ledger. On the other hand, nonprivate data require no encryption and are uploaded directly to IPFS. Every data access and storage is recorded in the blockchain, ensuring data security. To download data, user requesters are required to authenticate and authorize their access by checking the Id-block file associated with the requested data in the blockchain. Once authorization has been granted, the manager utilizes the data retrieved from the blockchain to retrieve the requested data information. As illustrated in Figure 8b, space–time big data download a process. Following privacy assessment, the manager proceeds to reconstruct the data into their original form through defragmentation and decryption. File scrambling data encoding is reduced from #001, #001, #010, #000, #110, #101, #111, and #100 to #001, #001, #010, #011, #100, #101, #101, and #111, ultimately obtaining the correct information. Nonprivate data are reconstructed and restored directly.

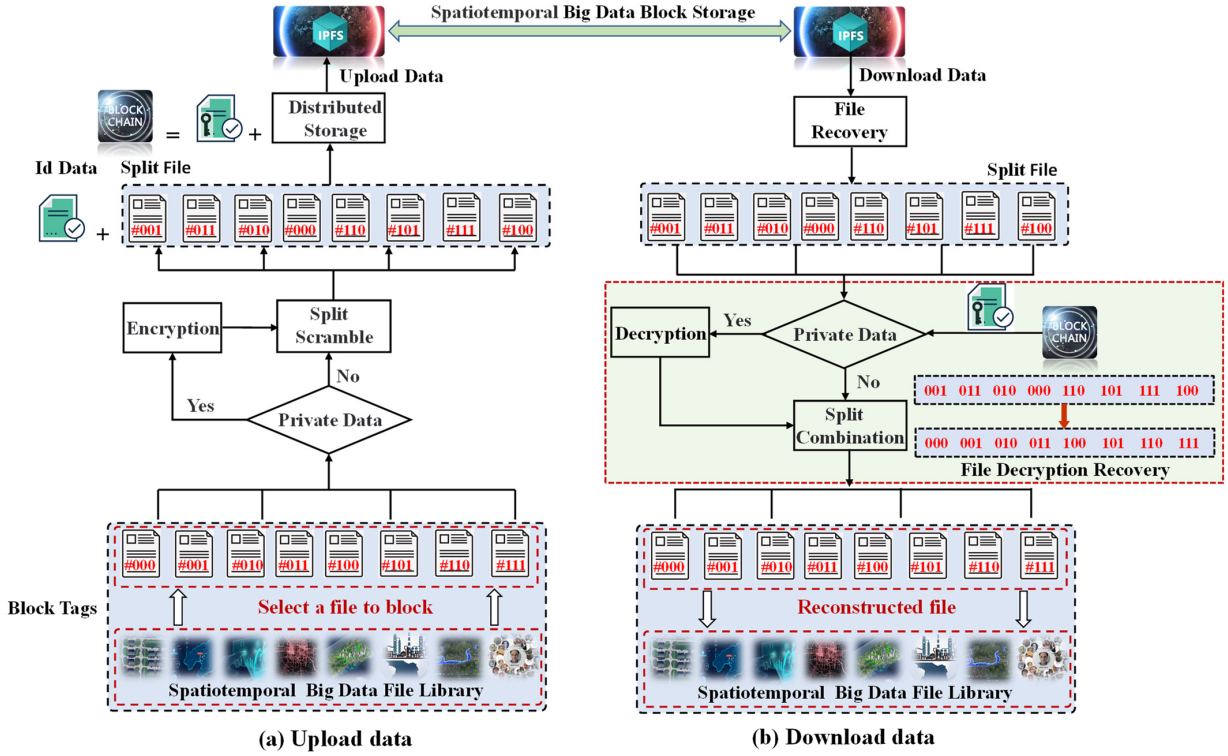

Figure 8. Spatiotemporal big data upload and download process.

The Id Data record the order of the ENCRYPT PART file, which is sent to IPFS. Figure 7 has been labeled SN, ON.

Upload data: First, select a file in the spatiotemporal big data file library and divide it into parts (#000-111). The second step is to encrypt the sensitive information of the judgment data. ENCRYPT the PART and record it as ENCRYPT PAR (#000-#111). In the third step, self-ordering, ENCRY PART (#001-#100), the previous order ENCRYPT PART (#000–#111) is stored in Id-Data.

Download data: First, restore the ENCRYPT PART (#001-#100) to the original order according to the order of the Id-Data ENCRYPT PART (#000-#111) in the blockchain. In the second step, the ENCRYPT PART (#000-#111) of this data is decrypted. The PART data are then combined into a file.

5. Simulation Result Analysis

The CSSoB technology entails the separation of sensitive and insensitive data during the storage process based on the user's preferences. The sensitive data are first divided into blocks, encrypted, and then stored in memory. Simultaneously, metadata are acquired. In contrast, insensitive data are directly separated into blocks and stored in memory, and metadata are also obtained. The metadata contain the block information of the data, which are then encrypted and stored in the blockchain. When retrieving data, the user accesses the block data by referring to the metadata information recorded in the blockchain. The data are then reconstructed to obtain the original data [16].

Compared with the literature [12], this scheme adopts the storage method of IPFS, which has the advantages of permanent storage of data, fast download of data, and security, and is not easy to be attacked. At the same time, the decentralized blockchain technology is adopted to make the data more secure and convenient to share. In the literature [13], the RSA encryption algorithm is adopted, while in this paper, the ECC algorithm is adopted, which has fast operation speed, low resource consumption, and the same encryption strength as the RSA algorithm, which can be achieved by using a shorter key.

As shown in Table 1, the performance of the ECC and RSA encryption algorithms is analyzed. It can be seen that in addition to low maturity, ECC has excellent performance in other aspects. Therefore, this paper adopts ECC to encrypt the data and ensure the safe storage of the data.

Table 1. Comparison of the ECC and RSA algorithms.

Name	Ripeness	Security (Depends on Key Length)	Operational Speed	Resource Consumption
RSA	High	Low	Slow	High
ECC	Low	High	Fast	Low

As shown in Table 2, the experimental hardware server of this system is Aliyun, with an 8-core CPU configuration, 16.00 GB memory, 2 TB hard disk size, and 500 Mbps LAN speed.

Table 2. Experimental hardware parameters.

Name	Data
CPU	8 core CPU
Memory	16.00 GB
Hard disk	2 TB
LAN speed	500 Mbps

As illustrated in Figure 9, the storage rate of CSSoB decreases gradually as the proportion of sensitive data increases. This can be attributed to the fact that CSSoB requires encryption of sensitive data, and as the number of sensitive data increases, the encryption time increases, thereby causing a decline in the overall system rate. Accordingly, it can be deduced that the proportion of sensitive data in the big data environment poses a chal-

lenge to the system's read and write performance, and thus, users must judiciously adopt different storage systems to manage their data according to their specific requirements.

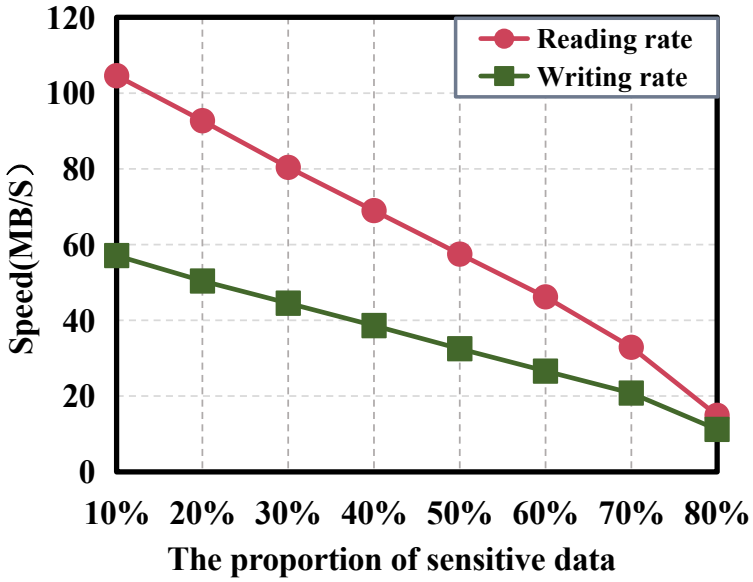

Figure 9. Sensitive data ratio vs. CSSoB performance test.

Figures 10 and 11 present a comparison of the read and write performance of the CSSoB scheme in the local and LAN settings. The test results show that whether it is a local or local area network, when the file is less than 40 MB, because the read and write rate is a linear rise, once the file is greater than 40 MB, the read and write rate will maintain a constant trend. This is because when the file is too small, the proportion of some initialization and network connection time increases greatly, resulting in a linear rise, and when the file is too large, the proportion decreases, and the processing data are more stable. The test also shows that the read/write rate of the CSSoB on the local is higher than that on the LAN. This is because the read/write rate of the CSSoB on the local is not affected by the network speed. The reason for the decrease in test reading in Figure 10 is that when the file size is large, the server needs to read the hard disk several times, which slows down the rate and thus decreases.

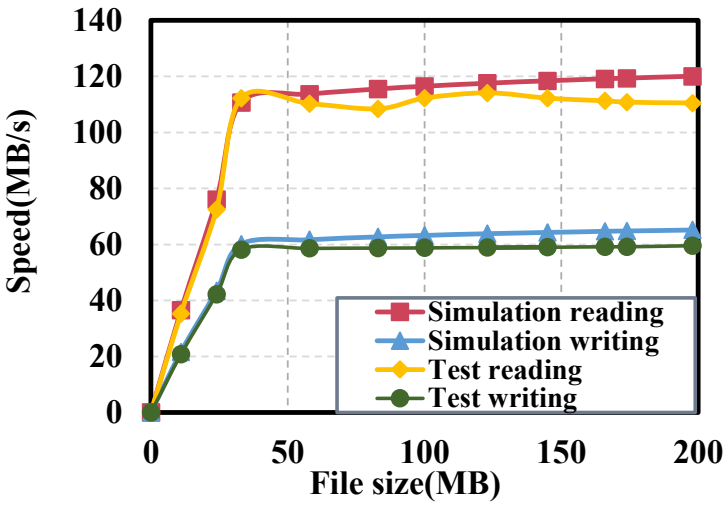

Figure 10. CSSoB performance local test.

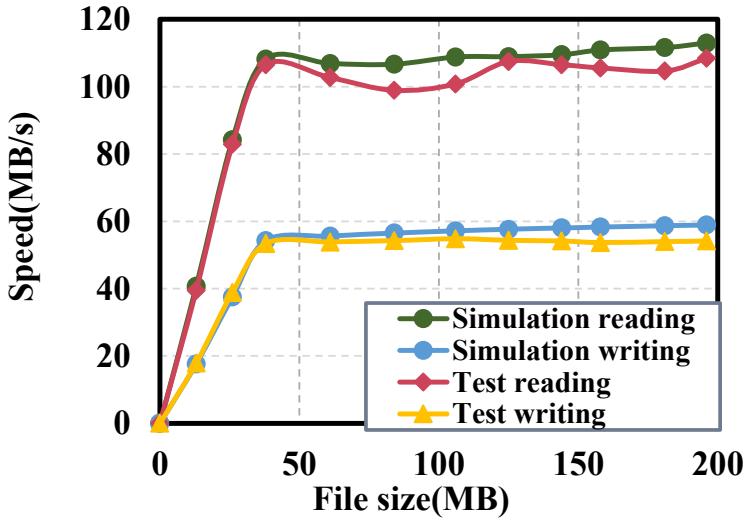

Figure 11. CSSoB performance LAN test.

As shown in Figure 12, in order to test the relationship between file size and latency, the read and write latency increases with the increase in file size, which is consistent with the actual situation. The write latency is larger than the read latency because the network bandwidth is limited or the server is busy.

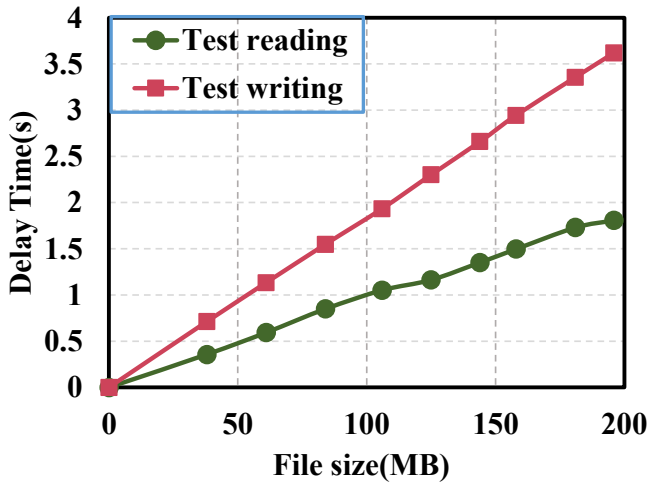

Figure 12. Performance testing for file size and latency.

The concept of TPS, or transactions per second, pertains to the capacity for transactional completion within a given time frame. In the domain of blockchain, TPS typically functions as a metric for evaluating a chain's processing prowess and throughput. A high TPS signifies the ability of the blockchain system to expeditiously process an extensive number of transactions, while a low TPS could result in transactional gridlock and latency. Consequently, enhancing TPS has emerged as a prominent objective pursued by many blockchain projects.

As illustrated in Figure 13, the TPS of CSSoB and HDFS are compared at different thread numbers. It is apparent from the figure that the throughput of CSSoB and HDFS increases nearly linearly with an increase in the number of threads. This phenomenon can be attributed to the fact that higher thread numbers translate to enhanced parallel processing ability, resulting in linear improvements in throughput. The CSSoB technical solution exhibits superior performance in comparison with the HDFS solution. When the number of threads is set to 100, the TPS of CSSoB surpasses that of HDFS by 7.8%.

Adoption of the CSSoB solution not only boosts storage performance, but also significantly enhances storage security.

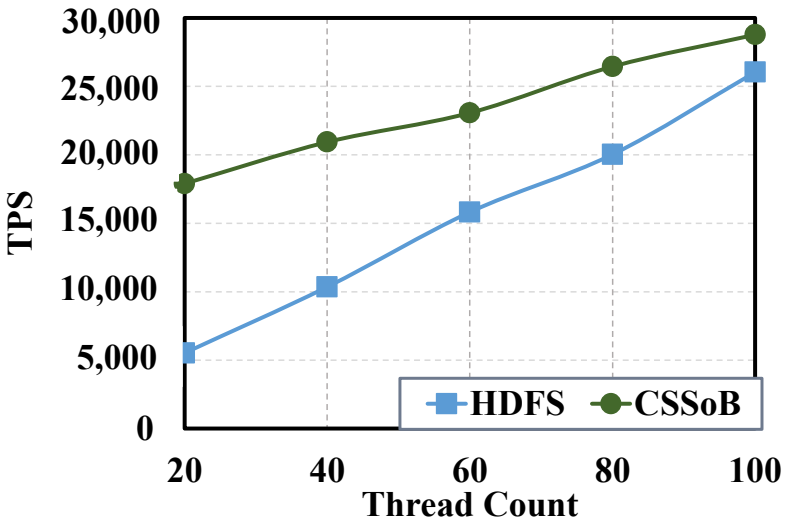

Figure 13. CSSoB and HDFS performance comparison TPS.

6. Conclusions

In view of the difficulties associated with the lack of transparency and ease of data tampering that exist in the present centralized storage methodology of spatiotemporal data, a spatiotemporal data-oriented blockchain is proposed, which integrates the decentralized, tamper-proof, and traceable characteristics of blockchain technology with spatiotemporal data management. Methods for constructing and querying the blockchain are also presented. To provide secure storage of spatiotemporal big data classification based on blockchain technology, a novel CSSoB scheme is proposed. The experimental results demonstrate that the storage rate of CSSoB gradually declines as the proportion of sensitive data increases. The performance of the CSSoB scheme is evaluated by comparing its read and write capabilities in both local and LAN environments. The results indicate that the absolute read and write speed is slower for smaller files, while larger files have greater speed. A comparison of the TPS between CSSoB and HDFS under various thread numbers reveals that when the thread number is 100, CSSoB outperforms HDFS by 7.8% in terms of TPS. Adopting the IPFS solution not only enhances storage performance, but also significantly bolsters storage security. These results suggest that the CSSoB scheme has substantial practical value. In the future, we will continue to study how to improve the efficiency of secure storage and improve the throughput of spatiotemporal big data secure storage.

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

Funding: National Natural Science Foundation of China (No. 41761081).

Informed Consent Statement: Informed consent has been obtained from all subjects for this study.

Data Availability Statement: The data used to support the findings of this study are available within the article.

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

References

1. Ren, Y.; Zhu, F.; Wang, J.; Sharma, P.K.; Ghosh, U. Novel vote scheme for decision-making feedback based on blockchain in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 1639–1648. [CrossRef]
2. Hagedorn, S.; Götze, P.; Sattler, K. The STARK framework for spatio-temporal data analytics on spark. In Proceedings of the Datenbanksysteme für Business, Technologies and Web (BTW 2017), Stuttgart, Germany, 6–10 May 2017; pp. 123–142.
3. Saggi, M.K.; Jain, S. A survey towards an integration of big data analytics to big insights for value-creation. *Inf. Process. Manag.* **2018**, *54*, 758–790. [CrossRef]
4. Baig, M.I.; Shuib, L.; Yadegaridehkordi, E. Big data adoption: State of the art and research challenges. *Inf. Process. Manag.* **2019**, *56*, 102095. [CrossRef]
5. Lv, D.; Zhu, S.; Xu, H.; Liu, R. A review of big data security and privacy protection technology. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 1082–1091. [CrossRef]
6. Nelson, B.; Olovsson, T. Security and privacy for big data: A systematic literature review. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 3693–3702. [CrossRef]
7. López-Alt, A.; Tromer, E.; Vaikuntanathan, V. On-the-fly multipartycomputation on the cloud via multikey fully homomorphic encryption. In Proceedings of the 44th Annual ACM Symposium on Theory of Computing, New York, NY, USA, 19–22 May 2012; pp. 1219–1234.
8. Zhou, Q.; Huang, H.; Zheng, Z.; Bian, J. Solutions to scalability of blockchain: A survey. *IEEE Access* **2020**, *8*, 16440–16455. [CrossRef]
9. Li, J.; Wu, J.; Jiang, G.; Srikanthan, T. Blockchain-based public auditing for big data in cloud storage. *Inf. Process. Manag.* **2020**, *57*, 102382. [CrossRef]
10. Alsamhi, S.H.; Lee, B. Blockchain-Empowered Multi-Robot Collaboration to Fight COVID-19 and Future Pandemics. *IEEE Access* **2021**, *9*, 44173–44197. [CrossRef]
11. Sahal, R.; Alsamhi, S.H.; Brown, K.N.; O'Shea, D.; McCarthy, C.; Guizani, M. Blockchain-Empowered Digital Twins Collaboration: Smart Transportation Use Case. *Machines* **2021**, *9*, 193. [CrossRef]
12. Alhazmi, H.E.; Eassa, F.E.; Sandokji, S.M. Towards big data security framework by leveraging fragmentation and blockchain technology. *IEEE Access* **2022**, *10*, 10768–10782. [CrossRef]
13. Ren, Y.; Huang, D.; Wang, W.; Yu, X. BSMD: A blockchain-based secure storage mechanism for big spatio-temporal data. *Future Gener. Comput. Syst.* **2023**, *138*, 328–338. [CrossRef]
14. Deebak, B.; Memon, F.H.; Dev, K.; Khowaja, S.A.; Wang, W.; Qureshi, N.M.F. TAB-SAPP: A trust-aware blockchain-based seamless authentication for massive IoT-enabled industrial applications. *IEEE Trans. Ind. Inf.* **2022**, *19*, 243–250. [CrossRef]
15. Wu, Q.; Han, Z.; Mohiuddin, G.; Ren, Y. Distributed timestamp mechanism based on verifiable delay functions. *Comput. Syst. Sci. Eng.* **2023**, *44*, 1633–1646. [CrossRef]
16. Alsulbi, K.A.; Khemakhem, M.A.; Basuhail, A.A.; Eassa, F.E.; Jambi, K.M.; Almarhabi, K.A. A Proposed Framework for Secure Data Storage in a Big Data Environment Based on Blockchain and Mobile Agent. *Symmetry* **2021**, *13*, 1990. [CrossRef]
17. Wang, W.; Xu, H.; Alazab, M.; Gadekallu, T.R.; Su, C. Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Trans. Ind. Inf.* **2021**, *18*, 1551–3203. [CrossRef]
18. Nguyen, G.T.; Kim, K. A survey about consensus algorithms used in Blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128.
19. Koblitz, N.; Menezes, A.; Vanstone, S. The State of Elliptic Curve Cryptography. *Des. Codes Cryptogr.* **2000**, *19*, 173–193. [CrossRef]
20. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
21. Tian, M. A scalar multiplication algorithm for wireless networks. *Shandong Sci.* **2009**, *22*, 5.
22. Zhu, F.; Qiu, L. Electronic document flow scheme based on Web service and elliptic curve. *Silicon Val.* **2010**, *14*, 2.
23. Hou, Z.; Li, L. Study on the overall algorithm design and optimization of Elliptic curve cryptographic system (ECC). *J. Electron. Sci.* **2004**, *32*, 1904–1906.
24. Kumar, D.S. Encryption of Data Using Elliptic Curve Over Finite Fields: Academy & Industry Research Collaboration Center (AIRCC). *Int. J. Distrib. Parallel Syst. (IJDPS)* **2012**, *3*, 3125. [CrossRef]
25. Song, C. Implementation of Elliptic Curve Cryptosystem Based on matlab. *J. Northwest Univ. Natl. Nat. Sci. Ed.* **2013**, *34*, 4.