applied sciences

MDPI

Article

Security Authentication Mechanism of Spatio-Temporal 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] West Anhui University, Luan 237000, China; 03000122@wxc.edu.cn
* Correspondence: 11301057@kust.edu.cn

Abstract: Spatiotemporal big data are a kind of data that marks time information and geographic location and has been widely applied in various fields. However, there are always security issues with spatiotemporal big data, especially in data collection and authentication. Traditional authentication protocols are less efficient in the face of ultra-large-scale IoT (Internet of Things, IoT) device verification, and the threat of single-point failure is relatively large. Given these complications, a group authentication scheme is proposed in this paper with blockchain spatiotemporal big data. The decentralization of the blockchain is utilized to solve the single point of failure, and the single-point authentication is combined with the group authentication, the authentication efficiency is improved through the group authentication, and the illegal nodes are accurately identified using the single-point authentication. The simulation results demonstrate that using the MHT (Merkel Hash Tree, MHT) algorithm for group authentication can effectively improve the authentication efficiency of the entire system when the number of users exceeds 200. The time overhead is only 4 ms when the number of users is 16,000. It can have a large throughput (400–500 tps) and a low latency (1–2 s) at the same time when the block size is 1500 KB. This study not only verifies the legitimacy of each device and protects the security of spatiotemporal big data, but also significantly reinforces the authentication efficiency compared with similar schemes.

Keywords: spatio-temporal big data; Internet of Things; blockchain; group authentication; MHT

Citation: Zhou, B.; Zhao, J.; Chen, G.; Yin, Y. Security Authentication Mechanism of Spatio-Temporal Big Data Based on Blockchain. *Appl. Sci.* **2023**, *13*, 6641. https://doi.org/10.3390/app13116641

Academic Editor: Yoshiyasu Takefuji

Received: 7 April 2023
Revised: 19 May 2023
Accepted: 22 May 2023
Published: 30 May 2023

1. Introduction

With the continuous development of Internet of Things devices and big data technology, massive spatiotemporal big data have been generated around the world. Spatiotemporal big data are a kind of big data, which are used to describe the associated movement of participants in space and time. Common spatio-temporal data types include mobile phone positioning data, shared commodity usage record data, and population migration data [1]. Despite the potential value of spatio-temporal big data, they also pose data security issues. Most traditional security authentication protocols adopt centralized authentication methods. Such authentication protocols [2–4] need to rely on trusted third parties, such as certificate authorities and authentication servers. Hence, there is a threat of a single point of failure. Most traditional authentication is the verification of a single device [5–9]. In the huge world of the Internet of Things, the efficiency of the verification of a single device is too low to satisfy current needs.

As a new decentralized distributed system technology, blockchain can be combined with traditional solutions in various scenarios to improve the security of the solution.

The literature [10] proposes a decentralized gateway architecture that utilizes a unique combination of public and private blockchain platforms through interoperability to connect private blockchains with end users, solving the problem that blockchains ultimately provide consumers with security and the difficulty of verifying services. In federated learning, the federated learning framework VFChain based on the blockchain system is proposed to realize verifiable and auditable federated learning and improve its security [11]. Blockchain technology is introduced into the domain name system [12] for providing safe and efficient DNS services. A blockchain-based decentralized authentication modeling scheme is proposed in the edge and IoT environment to realize a highly secure, highly reliable, and strong fault-tolerant authentication scheme [13]. An alternative authentication service method, DAuth, based on the Ethereum blockchain [14] is designed to achieve decentralized identity verification and lower the risk of user data being leaked or modified at will.

Blockchain, as a new decentralized distributed system technology, conforms to the distributed characteristics of spatiotemporal big data and provides a new method for solving spatiotemporal big data security authentication problems [15,16]. Nevertheless, the scheme implementation of blockchain-based spatiotemporal big data security authentication is in the exploratory stage, and many problems exist in blockchain-based methods. The current blockchain-related research on spatio-temporal big data security certification is mainly conducted from two perspectives: security architecture and security certification [7,17,18]. The security architecture focuses on the distributed characteristics of spatiotemporal big data authentication devices and how they better match the topology of the blockchain, so as to realize the unification of the logical structure of the two and enable the blockchain to better serve spatiotemporal big data safety certification. Concerning security authentication, peer-to-peer networks are constructed in the existing research mainly through gateway nodes, edge nodes, fog nodes, and other devices to support the deployment of blockchain, form a blockchain network, and realize spatiotemporal big data devices through blockchain.

This paper aims to wrestle with the above problems. Briefly, individual verification is combined with group authentication, group authentication is performed to improve authentication efficiency, and single authentication is adopted to accurately find illegal nodes. Among them, the identity authentication scheme of H ash is used to realize individual authentication; the blockchain, PUF technology, and MHT algorithm are employed to realize group authentication. The scheme can authenticate the data collection equipment to guarantee the legitimacy of each piece of equipment and finally protect the security of the blockchain spatiotemporal big data. The blockchain is utilized to store the ROOTHASH of the group, contributing to effectively lowering the single point of failure and lessening the risk of tampering. Moreover, the block size of 1500 KB can consider the system delay and system throughput.

2. Related Technologies

2.1. Spatio-Temporal Big Data Technology

The term spatio-temporal big data refers to massive data characterized by time and space information, and are generally collected by various physical and virtual sensors, covering multi-dimensional information such as time, geographical location, and environmental parameters. These data are multivariate, heterogeneous, high-dimensional, high-resolution, and achieve fast updates, with vital values and broad application prospects. Spatio-temporal big data are widely used in fields such as urban planning, traffic management, emergency rescue, environmental protection, weather forecasting, and natural resource management. Simultaneously, spatio-temporal big data raise issues of privacy and security, bringing about higher requirements regarding data collection, storage, analysis, and sharing [19]. Spatiotemporal big data are a kind of data that mark time information and geographic location and have been widely applied in various fields. However, there are always security issues with spatiotemporal big data, especially in data collection and

authentication. Traditional authentication protocols are less efficient in the face of ultra-large-scale IoT device verification, and the threat of single-point failure is relatively large. Given these complications, a group authentication scheme is proposed in this paper with blockchain spatiotemporal big data.

2.2. Blockchain Technology

Blockchain technology is a decentralized and distributed computing technology used to create traceable and non-tamperable data records. It is a successful basic technology of Bitcoin and has also been applied in various fields [20]. It is a network of nodes that maintain and update data records and link them using cryptography to form an unalterable distributed ledger, which guarantees data integrity and reliability. Each block contains a unique hash value calculated from the data stored in the block. Each time a new block is added to the chain, the new block also contains the hash value of the previous block. Therefore, it is impossible to modify the content of any previous block. Since the Root Hash value in this scheme is stored with the non-tamper able technology of the blockchain, the Root Hash of each group can be stored in the blockchain and cannot be tampered with for the purpose of tampering.

As illustrated in Figure 1, each block will have a unique "hash" value attached to it, calculated from the data stored in the block. Every time a new block is added to the chain, the new block also contains the hash of the previous block. Therefore, it is practically impossible to modify the contents of any previous block. In this scheme, the block stores the group identity information (org) and the corresponding group Root Hash (rth).

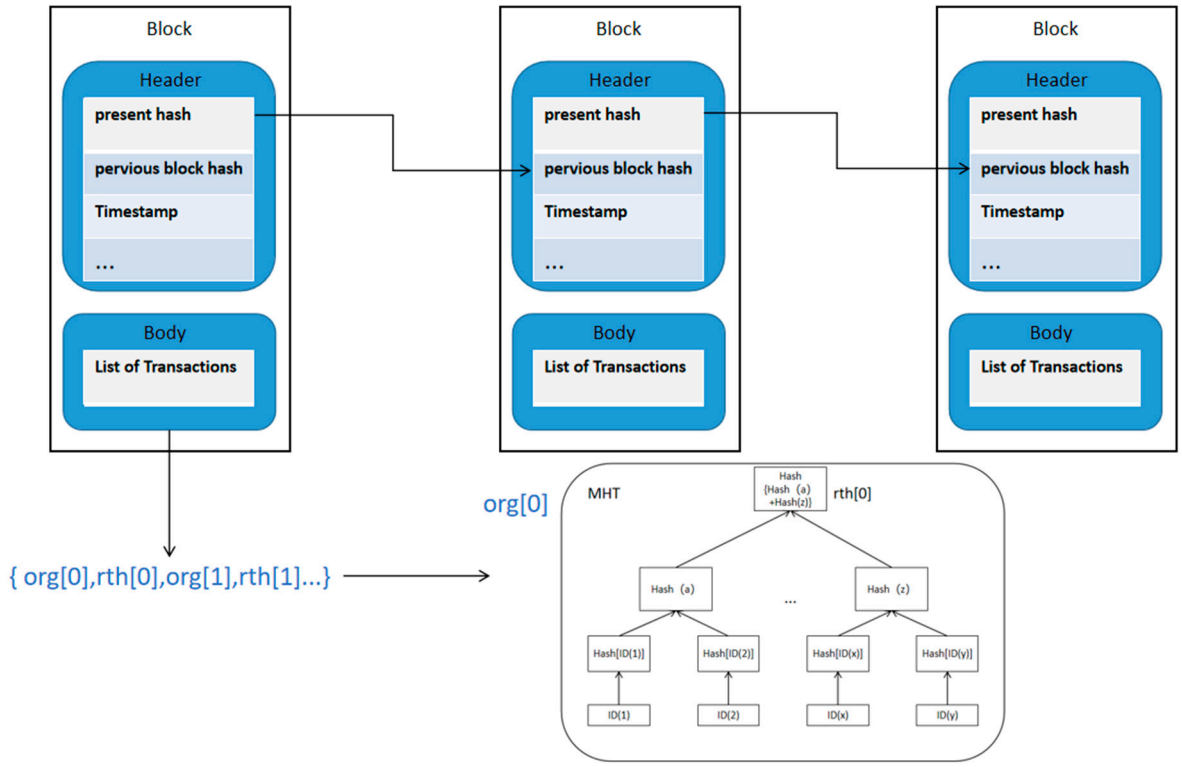

Figure 1. Blockchain structure diagram.

2.3. Physical Unclonable Technology

Physical Unclonable Function (PUF) is a security technology that utilizes inherent changes in hardware to generate an unclonable unique response [21]. Similar to human biometrics, PUFs can be employed for the unique identification of each integrated circuit chip. Each IC has different physical variations in path delay, transistor threshold voltage, and voltage gain owing to differences in silicon processing technology. In this system, each

IoT device integrates a PUF, which is adopted to generate the public identity of the device as a public key, realize message encryption, and achieve secure transmission through the PUF-based IoT security communication protocol. The system realizes the secure transmission of device IDs mainly via PUF technology to upload nodes and verification nodes. PUF technology can be used for the generation and storage of encryption keys because the keys generated by PUF technology are unique and cannot be copied, which noticeably enhances the security of the keys.

In this paper, a PUF is integrated into each IoT device; a public identifier is generated for each device using PUF technology and used as the public key of the device for message encryption; then, the PUF-based IoT security communication protocol realizes safe transmission. In this system, the PUF is mainly employed to securely transmit the device ID to the upload node and the verification node.

2.4. Merkel Hash Tree

The Merkle hash tree (MHT), which was proposed by Ralph Merkle in 1988 [22], is a tree data structure used to verify data integrity. The structure of the MHT is illustrated in Figure 2. In this solution, the unique identifier (i.e., device ID) of each IoT device is generated using PUF technology, and then the hash value of each ID is calculated as a leaf node through a hash algorithm (SHA256). Next, the adjacent hash values are combined, and the hash values are recalculated as their parent nodes until only one root node remains, namely, Root Hash. Root Hash represents the integrity and reliability of the entire IoT network, stored on the blockchain and maintained and updated by all nodes.

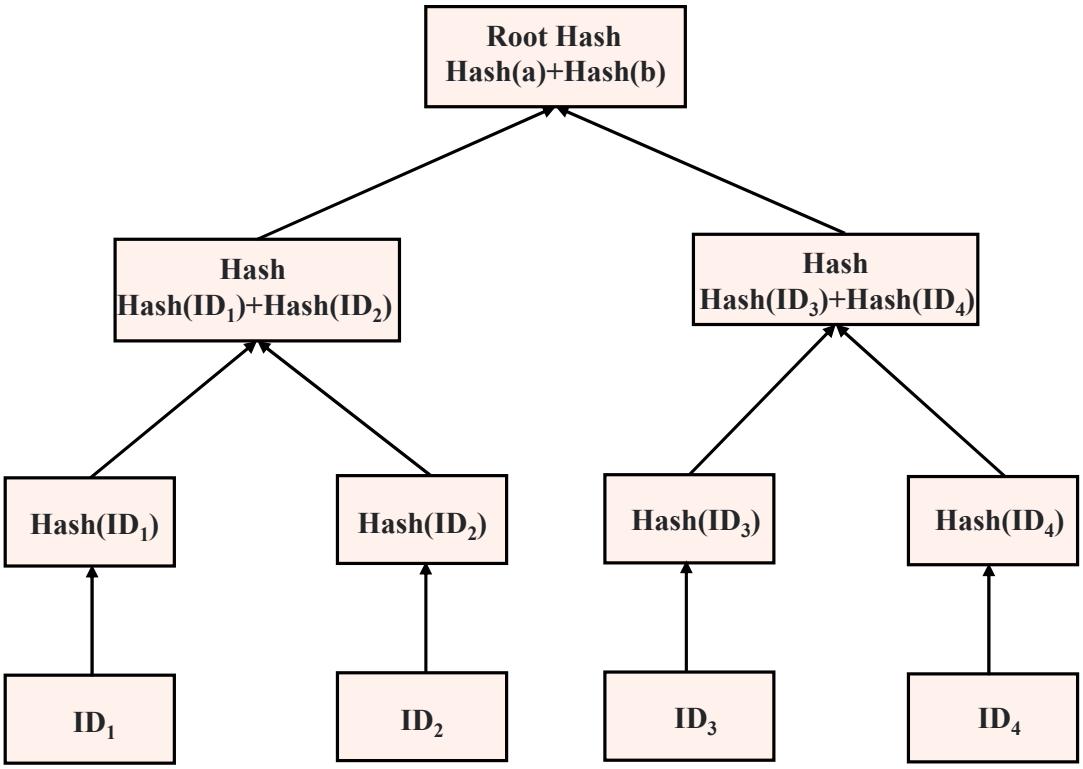

Figure 2. Merkel hash tree.

With the MHT, unique identifiers of IoT devices can be guaranteed against tampering and forgery. If the ID of any device changes, its hash value will change accordingly, resulting in changes in the entire MHT structure and Root Hash. In this way, any malicious attacker cannot alter the device ID without being detected, so as to guarantee the security and reliability of the IoT network.

Figure 3 exhibits the group authentication module of this paper. The MHT algorithm is mainly adopted in the upload node and the verification node, and the Root Hash of each device group is calculated by the ID of the space-time big data collection device. Root Hash (A) indicates the value obtained from all device IDs in the group stored in the initialization phase, and the default is a legal user. Root Hash (A′) reflects the value obtained from all device IDs in the group, and whether there are illegal users remains unclear. During the verification process, verification node B first receives all device IDs in group A to be verified and utilizes the MHT algorithm to obtain the Root Hash (A′) of the group to be verified. Afterward, the trusted Root Hash (A) in the blockchain is determined. Finally, Root Hash (A) and Root Hash (A′) are compared. If they are the same, the devices in the group are all legal. If they are different, there are illegal users in the group.

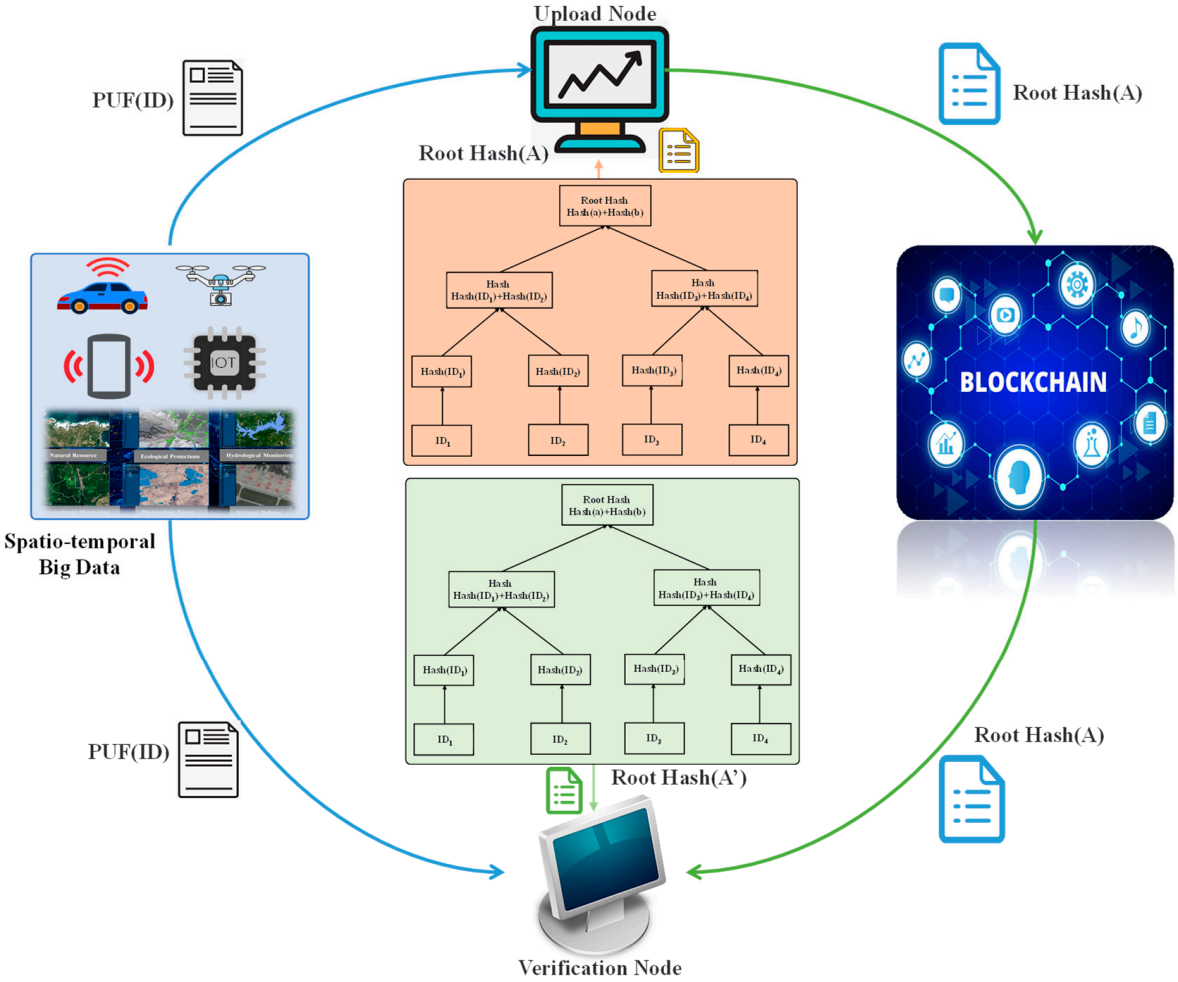

Figure 3. Blockchain authentication scheme.

3. Theoretical Analysis of Blockchain Identity Authentication

3.1. Principle of SHA256 Algorithm

Since the hash algorithm is deterministic, irreversible, anti-collision, and sensitive to input [23], it can better protect user identity and private key information. Hence, SHA256 in the hash algorithm is introduced into the authentication scheme in this paper. Its algorithm flow is detailed in Figure 4.

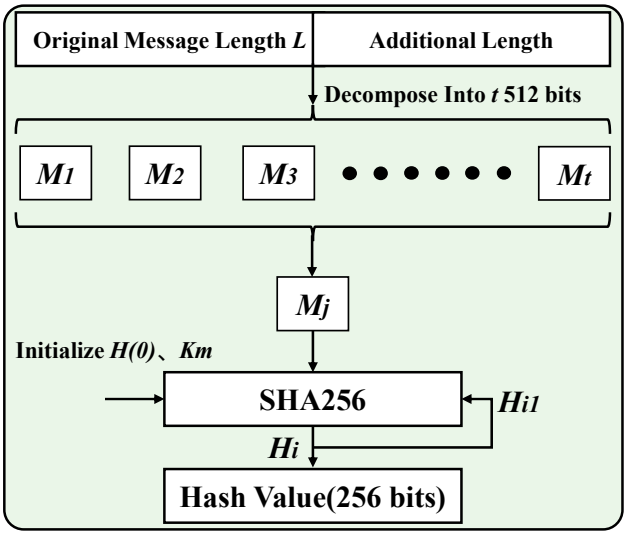

Figure 4. Flow chart of the SHA256 algorithm.

The SHA256 algorithm process is mainly divided into two stages: preprocessing and main loop. The message preprocessing stage completes the filling and expansion filling of the message. By adding one to the original message and adding enough zeros, the input original message is converted into t message blocks with a size of 512 bits. The SHA256 main function compresses each message block to obtain a hash value of 256 bits, where Hj can be updated iteratively.

3.2. Identity Authentication Based on Zero-Knowledge Proof

In 1989, MIT researcher Goldwasser and others proposed zero-knowledge proof [24], suggesting that the prover can prove that his statement is correct without disclosing any relevant information to the verifier. As revealed in Figure 5, zero-knowledge proof is adopted in a scenario where the group determines the illegal nodes, which can assure that group members complete the authentication without revealing their ID.

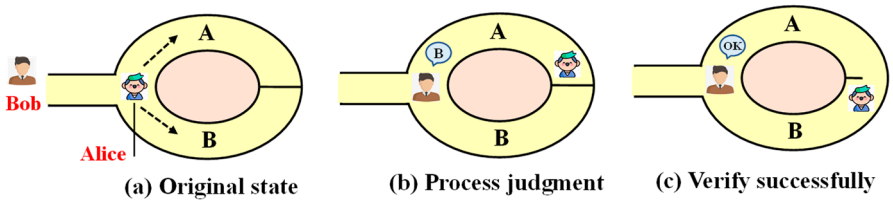

Figure 5. Zero-knowledge proof model.

Traditional identity authentication schemes may encounter attacks such as replay and impersonation during the authentication process, while zero-knowledge proofs can verify their identity without revealing their keys. Meanwhile, the verifier fails to obtain any valuable information about the keys. Given this feature, a new identity authentication scheme based on zero-knowledge proof is proposed in this paper in accordance with the authentication proposed in the literature [25–27]. The security of this scheme relies on the intractability of discrete logarithms. The scheme is expressed as:

$$x = lna^{y} + ev \ mod \ p \tag{1}$$

where p and q are two prime numbers, q denotes a prime factor of $p - 1$, a represents a factor of order q, and $a \neq 1$. Client H announces the authentication system parameters (p, q, a) for each node, and the upload node of each group registers its unique identity F for each node of the group. The upload node utilizes the hash function to generate a private key

following its identity. Additionally, N = hash(F||K), where K signifies the master key of the upload node.

As demonstrated in Figure 6, the specific process of the identity authentication protocol is detailed as follows.

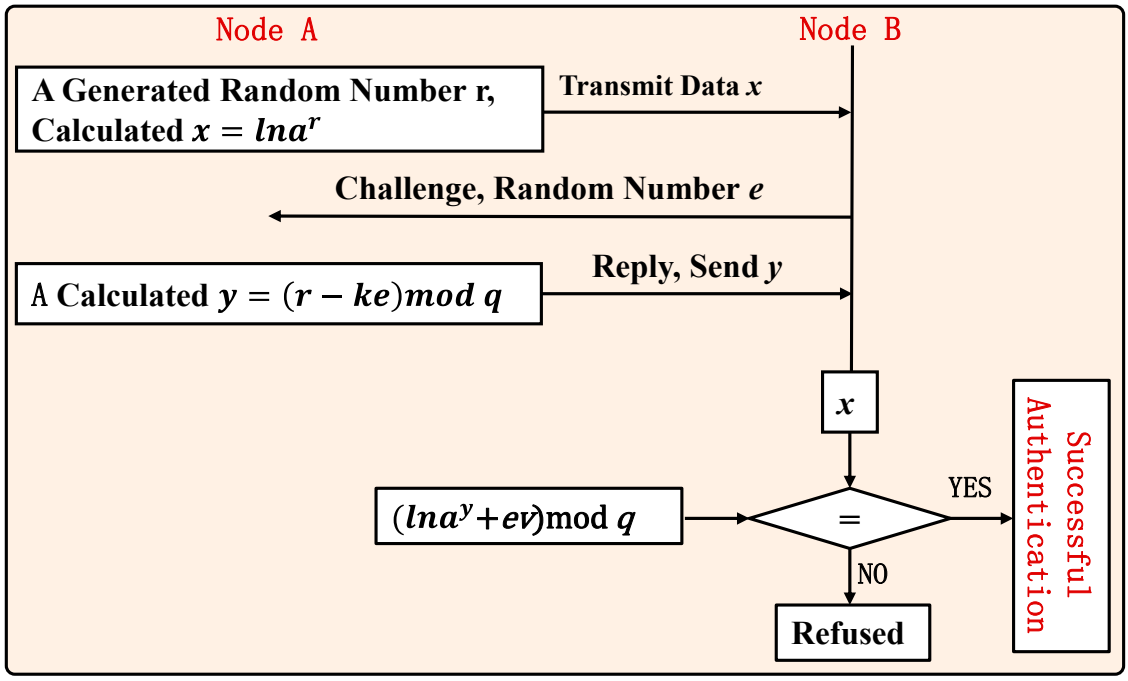

Figure 6. Identity authentication process.

Step 1: Node A requests to join the network, sends its own identity FA, uploads the node to generate a public key v = lnaˆk mod p (including the private key k = hash (F_A ||K)), and makes v public.

Step 2: Node A selects a random number r smaller than q, calculates it x = lnaˆr mod p, and makes x public. This process can be completed before both parties authenticate, saving authentication overhead.

Step 3: Suppose node B receives x and initiates a challenge to node A, and sends a random number e smaller than q to node A.

Step 4: Node A calculates y = (r − ke) mod q and sends y to node B.

Step 5: Node B verifies x = lnaˆy + evmod p and whether it is established. If it is established, the authentication of node A is successful; otherwise, the access of node A is rejected, and an authentication failure message is sent to the client.

4. Blockchain Group Authentication Scheme

4.1. Group Authentication Principles

In this scheme, a hybrid authentication method combining group authentication and single authentication is used for security authentication of spatio-temporal big data devices. In group authentication, the uploading node calculates the ID of all devices in the group with the MHT algorithm to obtain the Root Hash of the group and stores it on the blockchain. When the group device needs authentication, the group device transmits the ID to the verification node by using PUF technology. The verification node calculates the Root Hash of the group through the MHT algorithm. Whether the devices in the group are legitimate is determined by comparing the Root Hash with the Root Hash stored in the blockchain. If there are illegal devices in the group, the zero-knowledge proof is used for single authentication to accurately find the illegal nodes in the group. The experimental analysis suggests that the combination of group authentication and single authentication improves the overall authentication efficiency.

In actual data collection, the data-receiving end needs to authenticate the identity of each spatio-temporal big data authentication device to prevent illegal devices from transmitting wrong data to the data-receiving end. Therefore, the identity authentication of the device is imperative, which is directly associated with the accuracy of the collected data.

(1) Each device's ID is sufficiently random.

As displayed in Figure 7, there are three main components in this solution: group, blockchain, and client. Their functions are:

(2) Group: This is the unit of spatio-temporal big data collection, and the number of devices in each group can be specified by the user. Each group consists of several collection devices and an upload node with strong computing power. The role of the upload node is to receive the ID Hash of the collection device, build the MHT to obtain the Root Hash, and upload the Root Hash to the blockchain.

(3) Blockchain: As a decentralized security storage module, its main function in this system is to store the Root Hash of the group sent by the upload node in the group.

(4) Client: As a user of the system, it issues verification notifications and receives verification results.

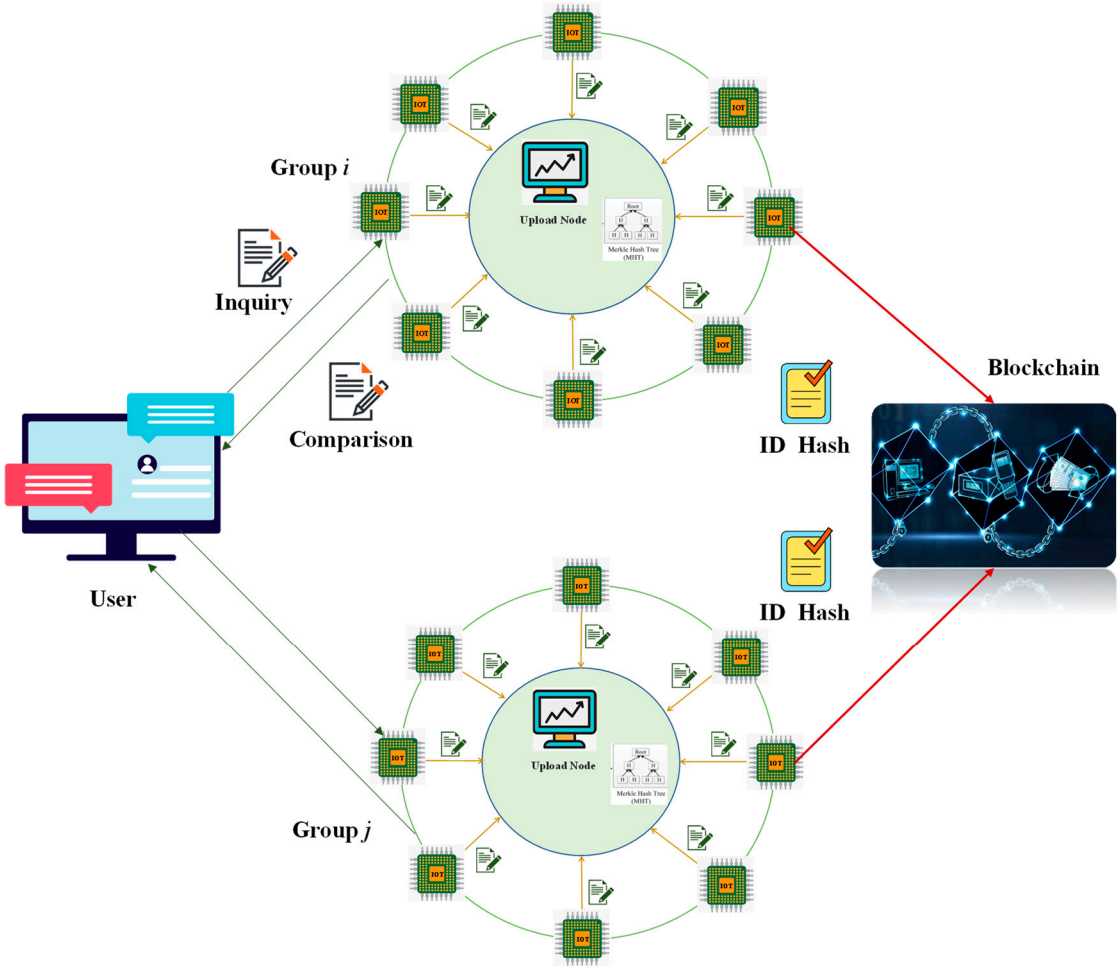

Figure 7. Blockchain-based spatio-temporal big data security authentication scheme.

This solution is divided into two phases: initialization phase and authentication phase. Table 1 presents the parameter list.

Table 1. Parameter annotation.

Parameter Name	Parameter Interpretation
CE	Client End
UN	Upload Node
VN	Verification Node
BLC	Blockchain
INPUT(x)	CE Selects a Group to Be Authenticated(X)
RANDOM(x)	Select The UN Of Group(Y) As VN(Random)
VERIFY(x)	CE Sent to Group X To Authenticated
AUTHORIZE(x,y)	CE Sent Authentication to Node Y
PUF(ID,x,y)	ID of a Transmission Group Member
QUERY(x)	Query The ROOTHASH(X) Stored in The Blockchain
ROOTHASH(x)	ROOTHASH Come From UN
ROOTHASH(x′)	ROOTHASH Come From VN
MHTupload [ID(x)] = ROOTHASH(x)	UN Obtains ROOTHASH(X) By Building MHT
MHTlocal [ID(x)] = ROOTHASH(x′)	VN Obtains ROOTHASH(X′) By Building MHT
COMPARE(x,x′)	Compare ROOTHASH(X) With ROOTHASH(X′)
PUF(ID)	Use PUF Technology to Encrypt Device's Ids
INTLALIZE(x)	Group X Is Initialized
INSERT[ROOTHASH(x)]	Save Group's Root hash(X) to the Blockchain
RETURN (A, B, Y/N)	The Result Message Sent by A To B
RETURN (A, B, Y)	The Knot Return Message Sent by A To B
ZKPV(x)	Zero-Knowledge Proof Algorithm

4.2. Initialization Phase

As reflected in Table 2, the main steps in the initialization stage are divided into three steps:

① The client sends an initialization command to group x.
② In group x, the device transmits the ID to the upload node, and the upload node performs the MHT algorithm on the ID of the device to the Root Hash value of the group.
③ The upload node saves the Root Hash of its group in the blockchain for preservation.

Table 2. Initialization.

CE	*UN*	*BLG*
INITLALIZE(x)	⟶ MHTupload[ID(x)] = ROOTHASH(x)	
	⟵ RETURN(x,CE,Y)	
		ROOTHASH(x) ⟶ INSERT[ROOTHASH(x)]
		⟵ RETURN(BLC,CE,Y/N)
IF Y → FINISH		
IF N → REPEAT		

4.3. Authentication Stage

Before the authentication phase starts, upload nodes of groups other than this group should be selected as authentication nodes to prevent collusion attacks.

The authentication phase is divided into eight steps (Figure 7):

① After receiving the verification challenge, all devices of group X use PUF technology to upload their IDs to the verification node (the upload node of group Y is randomly selected).
② After the verification node receives the ID Hash of the device in group X to be verified, it constructs the MHT to obtain the Root Hash (ROOTHASH(X′)) of the group to be verified.

③ The verification node queries the Root Hash (ROOTHASH(X)) of the group to be verified from the blockchain. Compare ROOTHASH(X) and ROOTHASH(X′). If they are the same, the devices to be verified in this group are legal; if not, it is necessary to verify whether there is any illegal device in group X.
④ Upload the verification result to the client.
⑤ If the verification finds an illegal user, the client sends a challenge to the upload node of group X to be verified.
⑥ The upload node of group X performs zero-knowledge proof identity authentication for the devices in the group.
⑦ The group X upload node uploads the authentication result to the client.
⑧ Based on the information provided by group X, the client finds out the information uploaded by illegal nodes and deletes it.

Table 3 provides a detailed flowchart of the verification phase. Figure 8 shows that the authentication of the system is initiated by the client. After the group receives the command, it will return a message to the client. If no return message is received within the set time, the client will resend the message to the group.

The client first selects the group to be verified and the verification node. The group to be verified uses PUF technology to pass the device ID in its own group to the verification node. After the verification node receives the verification request, it uses the MHT algorithm to obtain ROOTHASH(X′). The verification node looks up the ROOTHASH(X) of the group to be verified from the blockchain and compares ROOTHASH(X) and ROOTHASH(X′). If they are the same, the devices in the group to be verified are legal; if not, there are illegal devices in the group to be verified. The verification node uploads the verification result to the client. If the client receives Y, the verification is successful, indicating that all members of the group to be verified are legal; if it receives N, the verification fails, and the upload node needs to perform zero-knowledge Illegal device operations within the group. Find out the illegal device through the zero-knowledge proof algorithm, and then pass the device ID to the client. The client finds out the uploaded illegal device data and deletes it, and the verification is completed.

Table 3. Verification phase.

CE	*UN*	*VN*	*BL*
INPUT(x)			
RANDOM(y)			
	VERIFY(x) → PUF(ID)	PUF(ID,x,y) →	
	← RETURN(UN,CE,Y)		
	AUTHORIZE(x,y) →		
		← RETURN(y,CE,Y)	
		MHTlocal[ID(x)] = ROOTHASH(x′)	
QUERY(x) →			
			← ROOTHSAH (x)
		COMPARE(x, x′)	
		← RETURN(y,CE,Y/N)	
IF Y → FINISH			
IF N → ZKPV(X)			
	ZKPV(x) → ZKPV		
	← RETURN(x,ce,n)		
DELETE(x)			
INITLALIZE(x)			
FINISH			

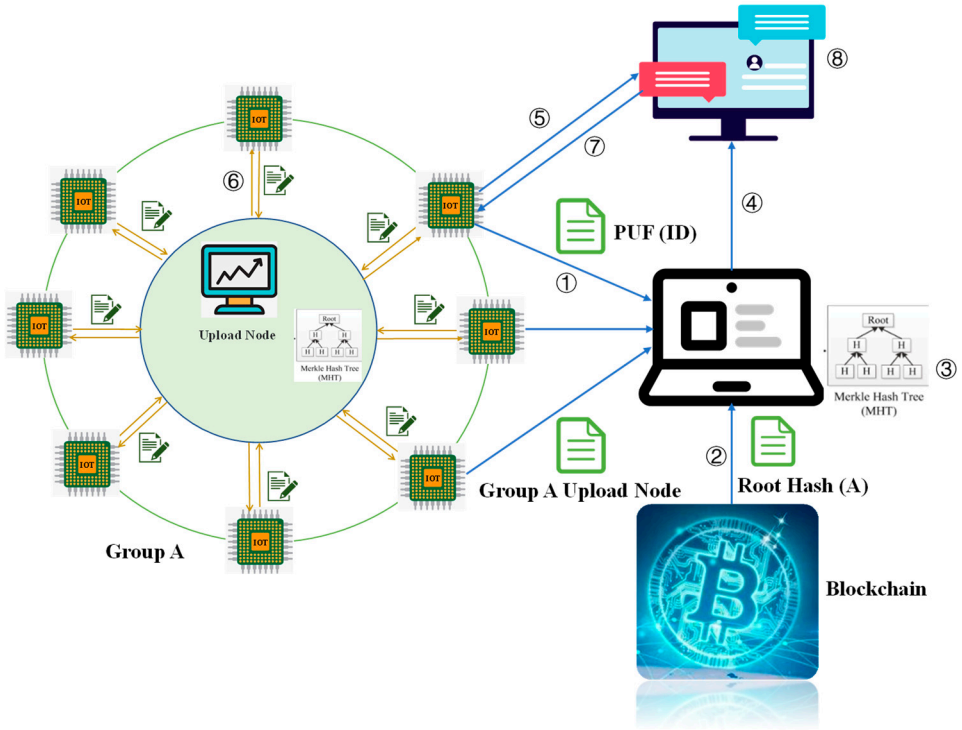

Figure 8. Authentication stage.

4.4. System Parameters

4.4.1. Experimental Hardware Parameters

As demonstrated in Table 4, the test hardware server of this system is Aliyun, the CPU is configured with eight cores, the memory is 16.00 GB, the hard disk size is 512 GB, and the operating system is Ubuntu16.4.

Table 4. Experimental hardware parameters.

Name	Data
CPU	8 core CPU
Memory	16.00 GB
Hard disk	512 GB
Cloud server	Aliyun

4.4.2. Data Structure Parameter

Figure 9 reflects that group authentication can be roughly divided into four layers: an initialization layer, authentication layer, comparison layer, and output layer. At the initialization layer, the initialization function is employed to save the group identity information and the corresponding rth (Root Hash) to the blockchain. The authentication layer adopts the authentication function to input the identity information of the group to be authenticated and the corresponding Present Root Hash (prth) into the system. In the comparison layer, the group identity information input in the authentication function is used to read the rth of the group and compare it with the prth. The output layer displays the results of the comparison layer. If the values are the same, authentication success is displayed. If the values are different, an authentication failure is displayed.

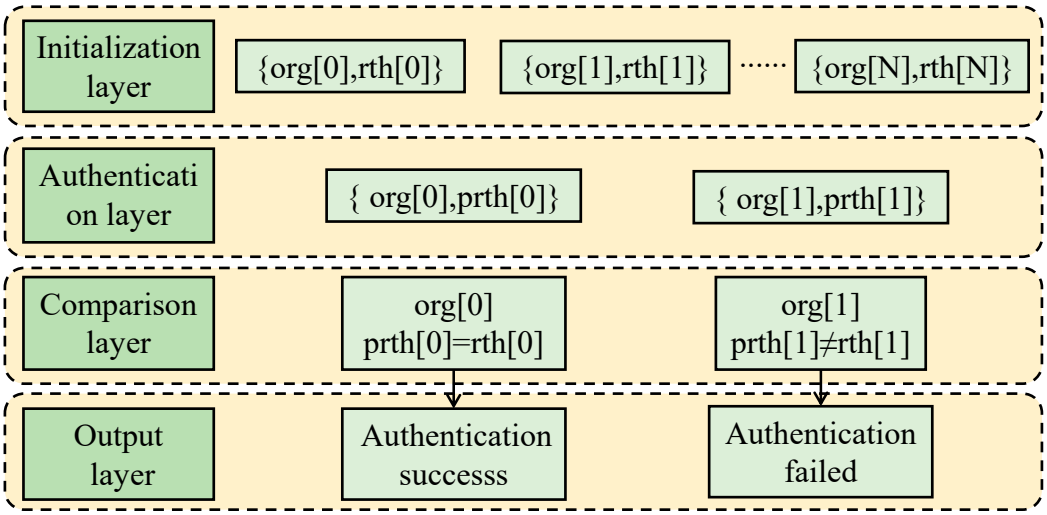

Figure 9. Group authentication structure diagram.

The Init function acts as the initialization function at the initialization layer. Each org (group) corresponds to a roothash (rth). The input to org is a 16-bit string representing the group ID, and the input to rth is a 256-bit string representing the roothash of the corresponding group stored in the blockchain. The Queryorg function is a query function at the authentication layer. org input is a 16-bit string representing the ID of the group to be queried. prth input is a 256-bit string representing the current roothash of the group to be queried.

5. Result Analysis

This paper explores the idea of an ID → Hash Value → Root Hash one-way transmission from left to right.

When the devices in the group transmit ID information in the network, this system uses PUF technology for secure transmission; it cannot be deciphered, even if it is intercepted; only the corresponding receiving node can convert the ciphertext into plaintext.

(1) The Root Hash stored in the blockchain cannot be changed nor tampered with.
(2) As long as there is an inconsistent leaf node in the MHT algorithm, the resulting Root Hash will be completely different. Thus, it is impossible to obtain the same Root Hash as the original ID to complete the authentication, though an attacker can find an ID from the Root Hash in the blockchain to the group to be verified.

As implied in Table 5, the authentication success rate for a group of 5000–10,000 members is tested through this experiment. The number of illegal members is 0, 5, and 20. Each group of experiments is performed 100 times. The verification success rate indicates that the group authentication of this scheme has strong security.

Table 5. Group authentication success rate experiment.

Number of Group Members	Number of Illegal Members	Number of Trials	Number of Successes	Number of Failures	Verification Success Rate
5000	0	100	100	0	100%
5000	5	100	100	0	100%
5000	20	100	100	0	100%
10,000	0	100	100	0	100%
10,000	5	100	100	0	100%
10,000	20	100	100	0	100%

5.1. Comparison of Different Authentication Schemes

Compared with the single authentication or group authentication used in the literature [14,27–30], this scheme uses mixed authentication, which combines single authentication with group authentication, uses group authentication to improve authentication efficiency, and adopts single authentication to find illegal nodes in the group (Table 6). At the same time, blockchain is introduced into the scheme to realize the decentralization of the authentication scheme and effectively solve the problem of a single point of failure. By introducing the MHT algorithm, the time cost is controlled at O(logn), which improves the authentication efficiency. The main contributions of this paper are described as follows. A group authentication scheme is proposed with blockchain spatiotemporal big data. The decentralization of the blockchain is performed to tackle the single point of failure. A single authentication is adopted to accurately find illegal nodes, and group authentication is combined with individual verification.

Table 6. Comparison of different authentication schemes.

Scheme	Single/Group/Mixed Authentication	Decentralization /Centralization	Time Cost	Certification Efficiency
Literature [27]	Single	Centralization	O (1)	Poor
Literature [28]	Single	Decentralization	O (1)	Poor
Literature [14]	Single	Decentralization	$O(n^2)$	Poor
Literature [29]	Group	Decentralization	$O(n^2)$	Median
This scheme	Mixed	Decentralization	O (logn)	Strong

5.2. Identity Authentication Security Performance Analysis

In this paper, an identity authentication scheme is designed for blockchain group authentication. Its security is based on the problem of discrete logarithms. The specific analysis is described as follows:

(1) The hash function has a single irreversible feature, which can ensure that the user's identity is not leaked.

(2) Due to the intractability of discrete logarithms, even if the attack node intercepts v and a, Eve cannot find k from Formula (2).

$$v = lna^k mod\ p \tag{2}$$

(3) The attacking node attempts to obtain r, and then intercepts y and e, so that he can obtain k from Formula (3); however, finding r from Formula (1) is also a problem that is difficult to solve for discrete logarithms. Similarly, the verification node cannot find r and then obtain k.

$$y = (r - ke) mod\ q \tag{3}$$

(4) Replay attack. The scheme is based on the challenge/response method, the authentication process is dynamic and interactive, and the random number sent for each authentication is different. Even if the attacking node obtains all the previously authenticated information, the validating node will not be fooled by outdated information.

(5) The verification node sends the message obtained during the transaction verification process to the attacking node. The verification node can only obtain v and y and judge whether the node is legal, and knows nothing about the relevant information of the private key k. Therefore, the security of the authentication process is guaranteed.

Legitimate users are only selected for authentication, as depicted in Figure 10. As the number of legitimate users increases, the time required for authentication is approximately positively correlated with the time of legitimate users. Therefore, the time cost consumed is huge when the number of authenticated users is very large.

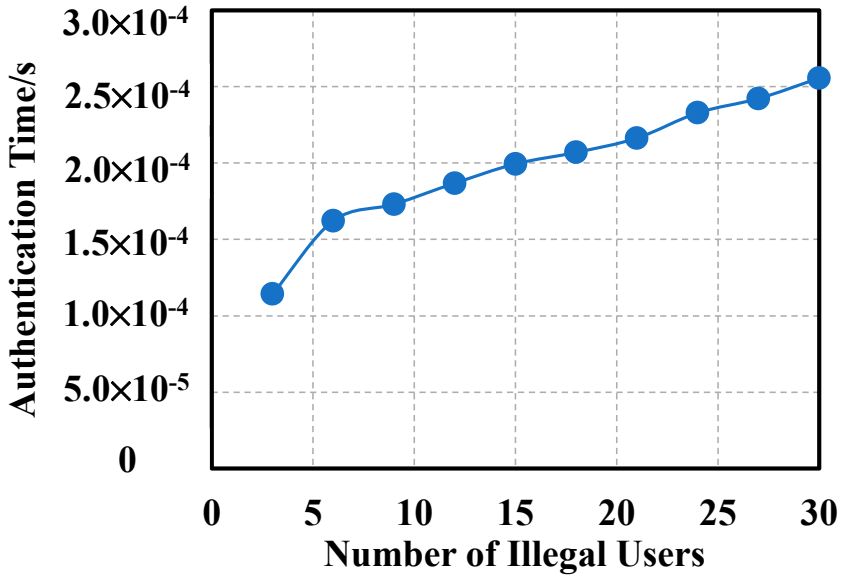

Figure 10. Invalid user authentication time.

Illegal users are only selected for authentication, as illustrated in Figure 11. As the number of illegal users increases, the time required for authentication is approximately proportional to the time of illegal users. Figures 8 and 9 suggest that under the same number of users, the authentication time is almost the same no matter whether it is illegal or legal users.

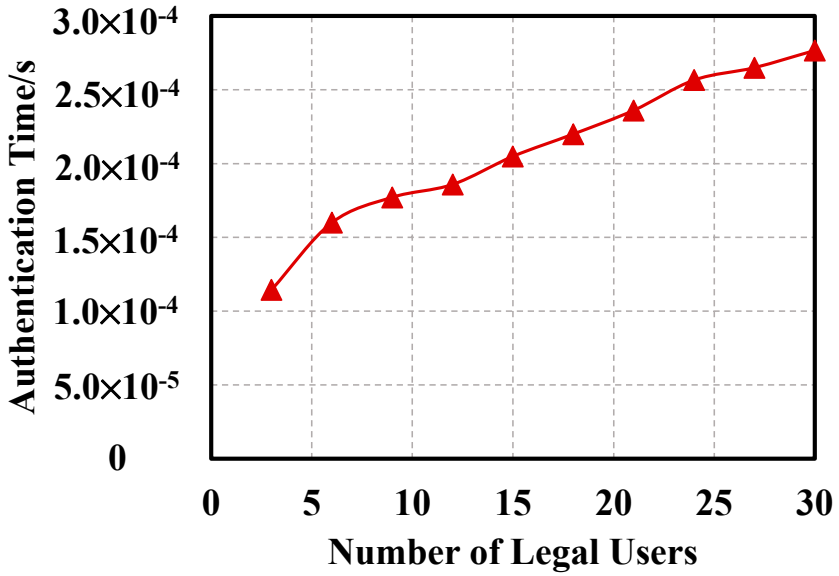

Figure 11. Valid user authentication time.

5.3. System Efficiency Analysis

This scheme theoretically depends on the computing power of the uploading node. The stronger the computing power of the upload node, the more members the group can have. The uploading node has to build the MHT within the specified time. Then, how many IDs can the uploading node complete the MHT construction within the specified time, and how many members can the group accommodate? In practical applications, the computing power of upload nodes can be reasonably allocated following the needs of users.

The time cost of the MHT algorithm and the zero-knowledge proof algorithm is proportional to the number of users. The time cost of the zero-knowledge proof algorithm

is lower than that of the MHT algorithm when the number of users is less than 200. The advantages of the MHT algorithm are gradually significant when the number of users is more than 200. The Merkel hash tree group authentication method can effectively improve system efficiency when IoT devices are connected on a large scale.

(1) Analysis of the internal efficiency of the system.

Considering that there are many kinds of Hash algorithms when constructing the MHT, the system efficiency of different algorithms is analyzed in this project. It is assumed in this simulated MHT algorithm that all users in the group are legal members.

Figure 12 illustrates the impact of different Hash algorithms on the MHT establishment time. The time cost of all algorithms is positively correlated with the number of users. The time cost of the SHA256 algorithm is lower than that of SHA384 and SHA512 under different numbers of users. The greater the number of users, the greater the gap between the time cost of the SHA256 algorithm and the other two algorithms. This is in accordance with the fact that the complexity of the SHA256 algorithm is lower than the other two algorithms, contributing to a lower time cost. Additionally, the simulation results are consistent with the theory. In other words, the Hash algorithm in the MHT algorithm in this system can employ SHA256 to improve the overall efficiency of the system.

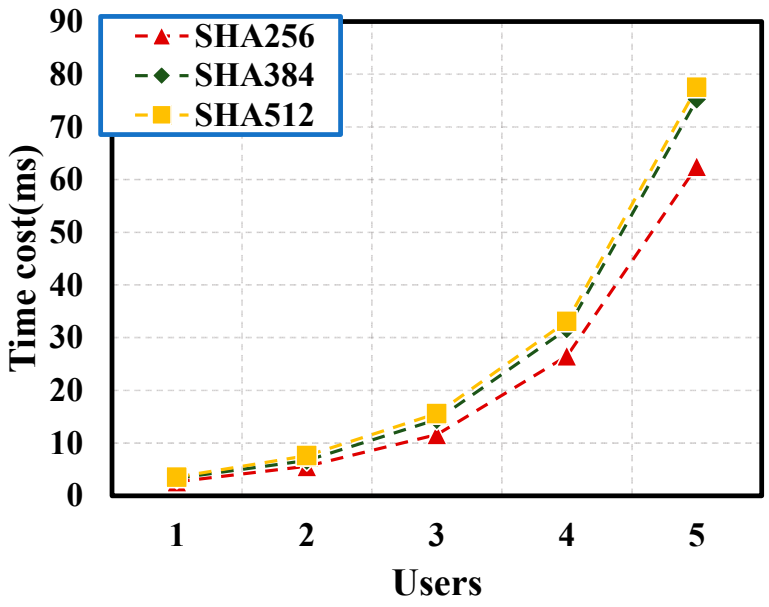

Figure 12. Authentication efficiency of different Hash algorithms.

(2) System throughput and delay analysis

In this study, transaction throughput (the number of transactions successfully delivered to the blockchain per second) and latency (the time difference between transaction delivery and response reception) are first tested. Two typical function-calling smart contract transactions are tested: (1) Insert, which changes the state of the blockchain and needs to be sent to peer nodes of other organizations for verification; (2) Verify, which can be directly executed.

As demonstrated in Figure 13, the throughput of the Verify function is proportional to the block size, consistent with the theory. When the number of nodes is set to three, the transaction size in the block is set to 4 KB, and the block generation time is set to 1 s. The relationship between the blockchain data block size and throughput is not a straight line. Nonetheless, the slope of the curve drops sharply when the block size is about 1500 KB. This is because the size of the network packet is set at about 1500 KB in the P2P network of the blockchain, requiring it to send the original block in packets after the block size exceeds

1500 KB. As a result, the transmission delay in the network presents a square increase, and the increase in data throughput decreases.

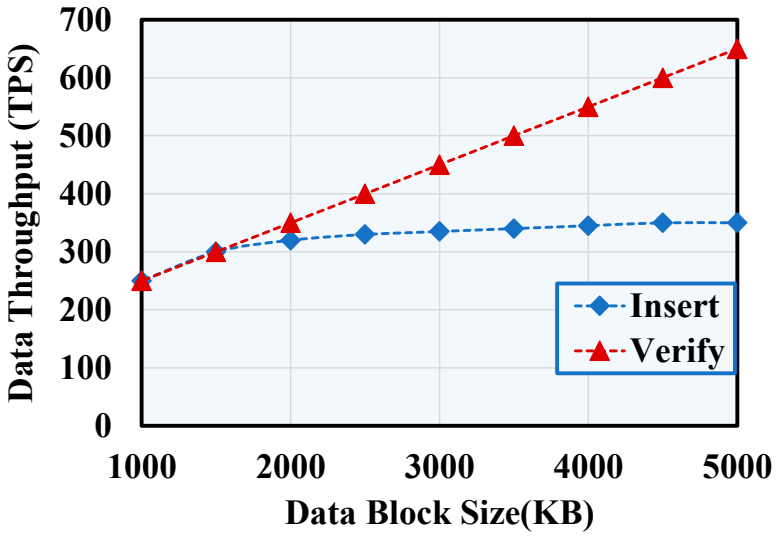

Figure 13. The relationship between data throughput and data block size.

When the block size is greater than 1500 KB, the transaction throughput using Insert increases slowly with the increase in the block size. The throughput is stable at 350 tps when the block size exceeds 1500 KB. However, the transaction throughput using Verify continues to increase as the block size increases.

Figure 14 unveils that the transaction latency using Verify does not change with the increase in the block size and remains around 10 ms, since such transactions should not be verified by other nodes. The increasing rate of transaction latency using Insert suddenly increases when the block size exceeds 1500 KB. Thus, if the block size is set to 1500 KB, the latency is about 1 s, and the throughput is 300 tps.

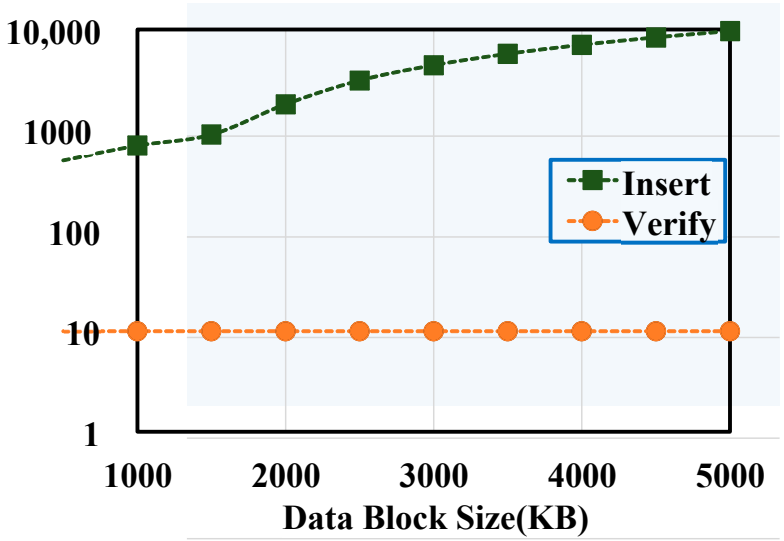

Figure 14. The relationship between data delay and data block size.

The system using Insert to write data in the blockchain during the initialization process should adopt Verify to verify the information during the verification process. Figures 13 and 14 suggest that the size of the block can be controlled at 1500 KB. At this time, the system can maintain low latency while obtaining a large throughput during the

initialization process and keep a good throughput during the verification process, satisfying the design requirements of the system.

6. Conclusions

This paper proposes a group authentication scheme based on blockchain spatio-temporal big data, in which the reference of blockchain technology enables the decentralization of this scheme, gets rid of the hidden danger of a single point of failure, and increases data security. The combination of authentication and single-point authentication enhances the efficiency of the solution. The analysis of the internal efficiency of the system reveals that when the number of users in the system is less than 200, the efficiency of ordinary single authentication is higher than that of group authentication, and the efficiency of the entire system is at a low level. Therefore, this solution is suitable for systems that connect IoT devices on a large scale. The larger the number of devices, the more noticeable the advantages of group authentication, and the higher the efficiency of the entire system. The model capacity parameters in the scheme can be flexibly adjusted according to user needs to help them find the most efficient configuration scheme.

Three different Hash value calculation methods in the MHT are compared in this paper to demonstrate that the SHA256 algorithm has a relatively low time cost. When there are 16,000 users, the time cost of the SHA256 algorithm is 62 ms, which is only 80% of the other two algorithms. The delay of Insert and Verify in the blockchain smart contract, the relationship between throughput and block size, and the optimal block size for the system to obtain greater throughput while maintaining low latency and block size. When the size is controlled at 1500 KB, it can take into account the high throughput (400–500 tps) and low latency (1–2 s) of the system. This scheme can not only verify the legitimacy of each device and protect the security of spatiotemporal big data, but also significantly reinforce the authentication efficiency compared with similar schemes.

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).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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.; 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]
2. Wu, F.; Li, X.; Sangaiah, A.K.; Xu, L.; Kumari, S.; Wu, L.; Shen, J. A lightweight and robust two factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *82*, 727–737. [CrossRef]
3. Ferrag, M.A.; Maglaras, L.A.; Janicke, H.; Jiang, J.; Shu, L. Authentication protocols for internet of things: A comprehensive survey. *Secur. Commun. Netw.* **2017**, *2017*, 6562953. [CrossRef]
4. Ferrag, M.A.; Maglaras, L.; Ahmim, A. Privacy-preservingschemes for ad hoc social networks: A survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 3015–3045. [CrossRef]
5. Braeken, A. PUF based authentication protocol for IoT. *Symmetry* **2018**, *10*, 352. [CrossRef]
6. Cui, Z.; Xue, F.; Zhang, S.; Cai, X.; Cao, Y.; Zhang, W.; Chen, J. A hybrid blockchain-based identity authentication scheme for multi-WSN. *IEEE Trans. Serv. Comput.* **2020**, *13*, 241–251. [CrossRef]
7. Hammi, M.T.; Hammi, B.; Bellot, P.; Serhrouchni, A. Bubbles of Trust: A decentralized blockchain-based auth-entication system for IoT. *Comput. Secur.* **2018**, *78*, 126–142. [CrossRef]

8. Kalra, S.; Sood, S.K. Secure authentication scheme for IoT and cloud servers. *Pervasive Mob. Comput.* **2015**, *24*, 210–223. [CrossRef]
9. Yang, H.J.; Li, Y.H. A Blockchain-Based Anonymous Authentication Scheme for Internet of Vehicles. *Procedia Comput. Sci.* **2022**, *201*, 413–420. [CrossRef]
10. Ghosh, B.C.; Bhartia, T.; Addya, S.K.; Chakraborty, S. Leveraging public-private blockchain interoperability for closed consortium interfacing. In Proceedings of the INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
11. Peng, Z.; Xu, J.; Chu, X.; Gao, S.; Yao, Y.; Gu, R.; Tang, Y. VFChain: Enabling Verifiable and Auditable Federated Learning via Blockchain Systems. *IEEE Trans. Netw. Sci. Eng.* **2021**, *9*, 173–186. [CrossRef]
12. Li, Z.; Gao, S.; Peng, Z.; Guo, S.; Yang, Y.; Xiao, B. B-DNS: A secure and efficient DNS based on the blockchain technology. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1674–1686. [CrossRef]
13. Ma, Z.; Meng, J.; Wang, J.; Shan, Z. Blockchain-based decentralized authentication modeling scheme in edge and IoT environment. *IEEE Internet Things J.* **2020**, *8*, 2116–2123.
14. Patel, S.; Sahoo, A.; Mohanta, B.K.; Panda, S.S.; Jena, D. DAuth: A decentralized web authentication system using Ethereum based blockchain. In Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019; pp. 1–5.
15. Biswas, S.; Sharif, K.; Li, F.; Nour, B.; Wang, Y. A scalable blockchain framework for secure transact-ions in IoT. *IEEE Internet Things J.* **2019**, *6*, 4650–4659. [CrossRef]
16. Huang, J.; Kong, L.; Chen, G.; Wu, M.Y.; Liu, X.; Zeng, P. To-wards secure industrial IoT: Blockchain system with credit-based consensus mechanism. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3680–3689. [CrossRef]
17. Almadhoun, R.; Kadadha, M.; Alhemeiri, M.; Alshehh, M.; Salah, K. A user authentication scheme of IoT devices using blockchain-enabled fog nodes. In Proceedings of the IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, Jordan, 28 October–1 November 2018. [CrossRef]
18. Bao, Z.; Shi, W.; He, D.; Chood, K.K.R. IoTChain: A three-tier blockchain-based IoT security architecture. *arXiv* **2018**, arXiv:1806.02008.
19. Dorri, A.; Jurdak, R.; Gauravaram, P. Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017. [CrossRef]
20. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 6 April 2023).
21. Chatterjee, U.; Chakraborty, R.S.; Mukhopadhyay, D. A PUF-based secure communication protocol for IoT. *ACM Trans. Embed. Comput. Syst.* **2017**, *16*, 1–25. [CrossRef]
22. Merkle, R.C. A certified digital signature. In *Advances in Cryptology—CRYPTO'89 Proceedings*; Springer: New York, NY, USA, 2001; pp. 218–238. [CrossRef]
23. Wang, J. Formalization of SHA256 Algorithm for Blockchain. Master's Thesis, Beijing University of Chemical Industry, Beijing, China, 2022. [CrossRef]
24. Li, W.H.; Zhang, Z.Y.; Zhou, Z.B.; Deng, Y. A Review of Concise Non-interactive zero-knowledge proof. *Acta Cryptologica Sin.* **2022**, *9*, 379–447. [CrossRef]
25. Zhou, X.C. Research on Authentication Mechanism Based on Zero-Knowledge Proof and Discrete Logarithm. Master's Thesis, Hefei University of Technology, Hefei, China, 2004. [CrossRef]
26. Gao, S.; Peng, Z.; Tan, F.; Zheng, Y.; Xiao, B. SymmeProof: Compact zero-knowledge argument for blockchain confidential transactions. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 2289–2301. [CrossRef]
27. Chen, Y.; Wang, X.L.; Ye, Q.; Jiang, H.H. Security Authentication Scheme Based on Zero-knowledge Proof. *Comput. Digit. Eng.* **2015**, *43*, 4. [CrossRef]
28. Sun, X.; Men, S.; Zhao, C.; Zhou, Z. A security authentication scheme in machine-to-machine home network service. *Secur. Commun. Netw.* **2015**, *8*, 2678–2686. [CrossRef]
29. Li, D.; Peng, W.; Deng, W.; Gai, F. A blockchain-based authentication and security mechanism for IoT. In Proceedings of the 27th International Conference on Computer Communication and Networks (ICCCN). In Proceedings of the 27th International Conference on Computer Communication and Networks (ICCCN), Hangzhou, China, 30 July–2 August 2018; pp. 1–6.
30. Shawky, M.A.; Jabbar, A.; Usman, M.; Imran, M.; Abbasi, Q.H.; Ansari, S.; Taha, A. Efficient Blockchain-based Group Key Distribution for Secure Authentication in VANETs. *IEEE Netw. Lett.* **2023**, *5*, 64–68. [CrossRef]