Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2020.012174
Article

Tech Science Press

Blockchain-Based Data Storage Mechanism for Industrial Internet of Things

Jin Wang[1,2], Wencheng Chen[1], Lei Wang[3], Yongjun Ren[4,*] and R. Simon Sherratt[5]

[1]School of Information Science and Engineering, Fujian University of Technology, Fuzhou, 350118, China
[2]School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410004, China
[3]School of Civil Engineering, Changsha University of Science & Technology, Changsha, 410000, China
[4]School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[5]Department of Biomedical Engineering, University of Reading, Earley, RG66AY, UK
[*]Corresponding Author: Yongjun Ren. Email: renyj100@126.com

Abstract: With the development of the Industrial Internet of Things and the continuous expansion of application scenarios, many development bottlenecks have followed. Its data security issue has become an obstacle to its widespread application. It has attracted substantial attention from both academia and industry. Blockchain technology has the characteristics of decentralization, openness and transparency and non-tampering. It has natural advantages in solving the security of the Industrial Internet of Things. Accordingly, this paper first analyzes the security risks associated with data storage in the Industrial Internet of Things and proposes the use of blockchain technology to ensure the secure storage of data in the Industrial Internet of Things. In traditional blockchains, the data layer uses Merkle hash trees to store data; however, the Merkle hash tree not able to provide non-member proof, which makes it unable to resist attacks from malicious nodes in the network. To solve this problem, this paper replaces the Merkle hash tree with a password accumulator to provide member proof and non-member proof. Moreover, the existing accumulators have trapdoors and cannot be updated in batches, and unable to meet the blockchain's expansion requirements. This paper presents an improved RSA accumulator and gives the definition of the accumulator. Finally, this paper uses RSA to construct a batch update accumulator scheme without trapdoor, and shows that the scheme is feasible through correctness and security.

Keywords: Industrial Internet of Things; blockchain; RSA accumulator; data storage

1 Introduction

With the rapid development of the Internet of Things technology, a series of national strategies, such as Made in China 2025, the American Advanced Manufacturing Partnership Program, and German Industry 4.0 have been proposed [1]. In this context, the Industrial Internet of Things has emerged at the historic moment and has become an important part of the intelligent transformation of the global industrial system

[2]. Medical system [3], the Internet of Vehicles (IoV) [4–6], artificial intelligence [7], dam security systems [8,9], etc. Underpinning all this is the acceleration of the industrial Internet of Things, rapid maturity, the arrival of the Industrial Internet of Things era has no doubt. The Industrial Internet of Things is to realize the flexible allocation of manufacturing raw materials, the on-demand execution of manufacturing processes, the reasonable optimization of manufacturing processes and the rapid adaptation of the manufacturing environment through the interconnection of industrial resources, data communication and system interoperability to achieve efficient use of resources [10,11]. In order to building a new industrial ecosystem which is driven by services.

With the integration of industrial Internet of Things technology and traditional industries, the Industrial IoT has profoundly changed the mode of production, organization and business model of traditional industries [12]. Traditional technology has been unable to meet the needs of the future industrial Internet of Things. Industrial Internet of Things projects are usually applied to enterprises in raw material procurement, inventory management, downstream sales and other links. Many of the information systems among these subjects are independent of each other, and there is a problem of data forgery [13,14]. In terms of saving user data, under the current centralized management mode of the Industrial Internet of Things, there is a possibility of data loss due to failure of individual devices [15]. When the amount of data information is too large, it will increase the burden of the central server. In terms of operation and maintenance costs, the current industrial IoT data streams are aggregated into a single central control system [16]. With the continuous evolution of low-power wide-area technology (LPWA) [17], the future industrial IoT equipment will grow geometrically and centralized service cost is unaffordable. In terms of data transmission, due to the open nature of the wireless network, the nodes lacking security are very vulnerable, along with the wireless information transmitted between the devices is vulnerable to threats [18–20]. The possible threats are mainly spread of spam data, DDoS attacks and cross heterogeneous network attacks. Therefore, the security issues of the Industrial Internet of Things have attracted much attention, and the blockchain can provide the best solution.

In order to improve the data security of the Industrial Internet of Things, some scholars have integrated blockchain technology into the Industrial Internet of Things to improve security [21–23]. With the characteristics of decentralization, openness, transparency and non-tampering, blockchain technology provides trust, transparency and secure data guarantee for the Industrial Internet of Things. Moreover, encrypted contracts between industrial IoT devices can be recorded as smart contracts on the blockchain and automatically executed to improve efficiency. In blockchain technology, the data layer of the blockchain uses a Merkle tree to store data. However, the Merkle tree stores data has the following shortcomings: It can only provide member proof, cannot provide non-member proof, the storage takes up large memory, and members cannot be deleted at will. In recent years, cryptographic accumulators, a potential alternative to Merkle trees for blockchains, have attracted more and more interest. Because the accumulator has the characteristics of strongness, universality, compactness, it can provide the advantages of non-member proof, delete members at will, reduce data storage memory, so this paper introduces the accumulator in the blockchain and replaces it with the accumulator Merkle tree can reduce the memory of node data storage, and also better protect privacy.

This paper studies the storage mechanism of blockchain data based on accumulators, and further proposes a method for secure storage of industrial Internet of things data based on blockchain. In addition, this paper presents the concept of batch update accumulator without trapdoors, uses RSA to construct a specific scheme, and demonstrates its security.

The remainder of this paper is structured as follows. Section 2 presents some related work about Industrial Internet of Things, blockchain technology, prime representation, RSA accumulator, strong RSA assumption and Batch and aggregation. In Section 3, the data security issues of the Industrial Internet of

Things are described. In Section 4, moreover, the secure storage of data based on the blockchain-based Industrial Internet of Things is described. Finally, the conclusion is provided in Section 5.

2 Related Work

In this section, the industrial Internet of Things, blockchain technology, prime representation, batch and aggregation, RSA accumulator and strong RSA assumption are described.

2.1 Industrial Internet of Things

A typical IoT system has three levels [24]. One is the perception layer, which uses RFID, sensors, QR codes, etc. to obtain object information anytime and anywhere; the second is the network layer, through the integration of telecommunications networks and the Internet, the object information is accurately transmitted in real time; the third is the application layer, which the information obtained by the perception layer is processed to realize practical applications such as intelligent identification, positioning, tracking, monitoring and management [25]. The architecture is as illustrated in Fig. 1 below.

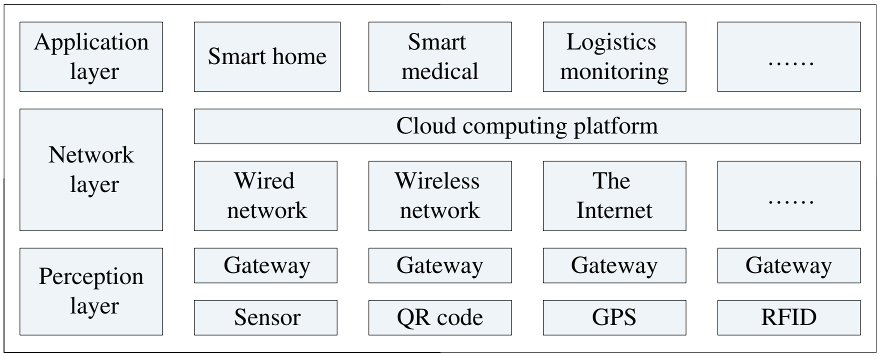

Figure 1: Internet of Things architecture

Compared with the traditional IoT architecture, the industrial IoT system architecture adds on-site management. The role of the on-site management layer is similar to an application sublayer, which can preprocess data at a lower level, and is an indispensable layer for real-time control, real-time alarm and real-time data recording in industrial applications.

Perception layer: The perception layer is composed of field equipment and control equipment, mainly for the perception of industrial machine information and the issuance of control instructions. Field devices mainly include temperature sensors, humidity sensors, pressure sensors, RFID, electric valves, transmitters, etc. These devices are directly connected to industrial machines and serve as a peripheral mechanism for sensing the control process. Control equipment mainly refers to PLC and other controllers. In industrial systems, PLC and other controllers are used to achieve lower-level high-speed real-time control functions, which are particularly important for industrial control. The control device and field device form a field bus control network, such as the commonly used CAN bus network, Profibus bus network.

On-site management: On-site management mainly refers to the local dispatch management center of the factory, namely the SCADA system. The dispatch management center acts as the local manager of the industrial system and the provider of the external interface of the industrial data, generally includes industrial database servers, monitoring servers, file servers and Web network servers. As a layer different from the traditional IoT system architecture, the on-site management layer plays an important role in the industrial IoT system. On-site management integrates the existing industrial monitoring system. Its existence enables some key industrial data from the perception layer to be recorded and processed in a timely manner. For some lower-level process control instructions that require real-time performance, it can quickly respond and make control decisions in a timely manner.

Network layer: The network layer uses telecommunications network or Ethernet to set up a transmission channel for the local data of the factory and the remote data analysis center, so that the data can be transmitted anytime and anywhere.

Application layer: The application layer is the ultimate embodiment of the Industrial Internet of Things. The application layer meets the needs of industrial applications and is deeply integrated with industry expertise. It uses big data processing technology to analyze the data from the perception layer, which mainly includes monitoring the production process, tracking and recording the operating status of industrial machines. It produces results that have guiding significance for the development of enterprises and industries, such as optimizing production processes, knowing production management, improving operating efficiency, and predicting industry development, to achieve a wide range of intelligence. Different companies can share the analysis and processing results of big data with each other, which plays a great role in promoting collaborative production among enterprises, optimizing social industrial structure, and improving overall social productivity.

2.2 Blockchain Technology

Blockchain is mostly known as the technology underlying the cryptocurrency Bitcoin [26]. The core idea of a blockchain is decentralization. This means that blockchain does not store any of its database in a central location, but will copy and distribute the blockchain on the participant network. Whenever a new block is added to the blockchain, every computer on the network updates its blockchain to reflect the change. This decentralized architecture has the advantages of tamper resistance and no single-point failure vulnerabilities, which can ensure robust and secure operation on the blockchain. The general concept on how the blockchain operates is presented in Fig. 2.

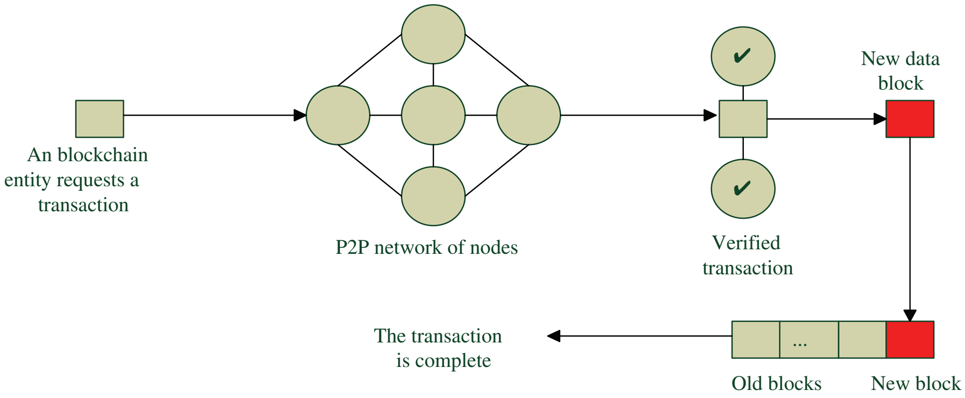

Figure 2: The concept of blockchain operation

Main components of Blockchain: Data block: Blockchain is essentially a block chain, a linear structure, starting from the so-called genesis block, all the way to each new block linked in the chain. Each block contains a number of transactions and is linked to its immediately-previous block through a hash label. In this way, all blocks in the chain can be traced back to the previous one, and no modification or alternation to block data is possible. In particular, a typical data block structure includes two main components, including transaction records and a blockchain header [26]. Here, transaction records are organized in a Merkle tree-based structure where a leaf node represents a transaction of a blockchain user. For example, a user may request to have associated metadata to establish a transaction that is also signed with the private key of user for trust guarantees. At the same time, the block header contains the following information: (1) Hash of the block for verification, (2) Merkle root, used to store a set of transactions in each block (3) Nonce value which is a number that is generated by consensus process to produce a hash value below a target difficulty level, and (4) Timestamp which refers to the time when the block was created. A typical blockchain structure is presented in Fig. 3.

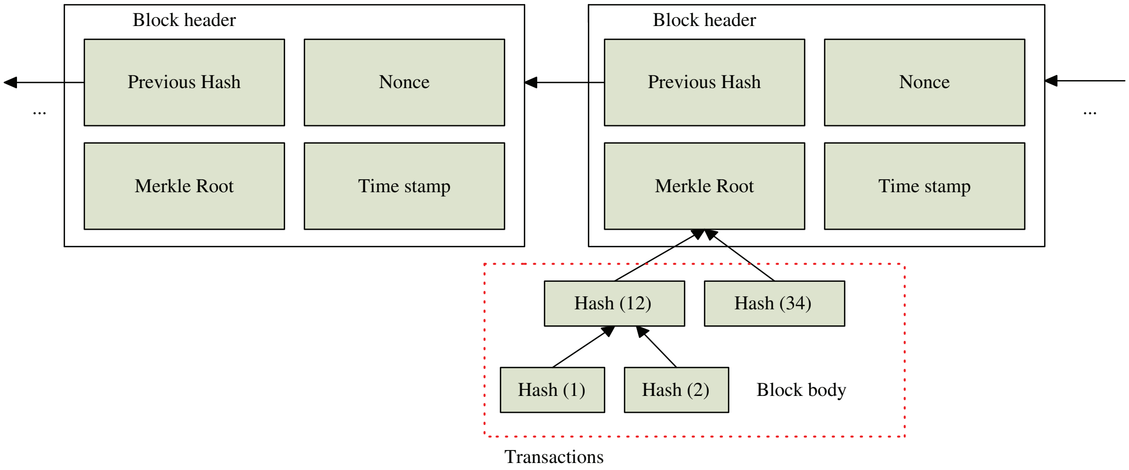

Figure 3: The data block structure

Distributed ledger (database): A distributed ledger is a type of database which is shared and replicated among entities on a peer-to-peer network. The shared database is available for all network participants within the blockchain ecosystem. Participants of the network can achieve on the agreement by a consensus mechanism in a distributed environment that does not require third parties to perform transactions.

Consensus algorithms: When nodes start to share or exchange data on the blockchain platform, there is no centralized parties to regulate transaction rules and protect data from security threats. In this regard, it is vitally to verify the trustworthiness of the block, track the data flow and guarantee a secure exchange of information to avoid fraud problems, such as double-spending attacks. These requirements can be met by using a verification protocol called a consensus algorithm. In the blockchain context, a consensus algorithm is a process used to reach agreement on a single data block among multiple unreliable nodes. An example of a consensus application is the Bitcoin blockchain. Bitcoin uses a proof of work algorithm (PoW) [27,28] as a consensus mechanism run by miners to ensure security in untrusted networks.

Smart contracts: Smart contracts are programmable applications that run on the blockchain network. Since the first smart contract platform known as Ethereum [29] was released in 2015, smart contracts have gradually become one of the most innovative topics in the blockchain field. For example, when a person signs a smart contract to transfer his funds, the funds will be automatically transferred through the blockchain network. Then the transfer information will be recorded as a transaction and stored on the blockchain as an immutable ledger [30]. This self-executing protocol that relies on code makes the smart contract immutable and resistant to external attacks [31–33].

2.3 Prime Representatives

For reasons of security and correctness, in the construction of this paper, the main representative concepts of the widely used elements will be quickly clarified. Initially introduced in [34], prime representatives [35] provide a solution whenever it is necessary to map general elements to prime numbers. In particular, one can map a k-bit element e_i to a $3k$ -bit prime x_i using two-universal hash functions.

This paper uses two general functions. $h(x) = Fx$, where F is a $k \times 3k$ Boolean matrix. Since the linear system $h(x) = Fx$ has multiple solutions, one k-bit element is mapped to more than one $3k$-bit elements.

Let H be a two-universal family of functions mapping $\{0,1\}^{3k}$ to $\{0,1\}^{k}$ and let $h \in H$. For any element $e_i \in \{0,1\}^{k}$, the prime number $x_i \in \{0,1\}^{3k}$ can be computed by sampling $O(k^2)$ times with high probability from a set of inverse $h^{-1}(e_i)$, so that $h(x_i) = e_i$.

2.4 RSA Accumulator

Suppose there is a set of k-bit elements $X = \{x_1, x_2, \ldots, x_n\}$. Let N be a k'-bit RSA modulus with $k' > 3k$, namely $N = pq$, where p, q are strong primes numbers. Using the RSA accumulator [36], we can represent X compactly and securely with an accumulation value $acc(X)$, which is a k'-bit integer defined as $acc(X) = g^{r(x_1)r(x_2)\ldots r(x_n)}\ mod\ N$, Where $g \in QR_N$ and $r(x_i)$ is a $3k$-bit prime representative, computed using a universal hash function h.

According to the accumulative value $acc(X)$, each element in the set X hash a member witness, the value is: $W_{x_i} = g^{\prod_{x_j \in X : x_j \neq x_i} r(x_j)}\ mod\ N$. Given the accumulated value $acc(X)$ and the witness W_{x_i}, you can verify the membership of x_i in X by computing $W_{x_i}^{r(x_i)}\ mod\ N$ and checking that it is equal to $acc(X)$. Any adversary A, who does not know $\varnothing(N)$, subject to computation restrictions, cannot find another set of elements $X' \neq X$ such that $acc(X') = acc(X)$ unless A breaks the strong RSA assumption.

2.5 Strong RSA Assumption

Given an RSA modulus N and a random element $x \in Z_N$, that is difficult (i.e., it can be done with probability that is $neg(k)$, which is negligible in the security parameter k) for a computationally bounded adversary A to find $y > 1$ and a such that $a^y = x\ mon\ N$.

2.6 Batch and Aggregation

This paper uses batch processing to describe a single operation corresponding to n items, rather than one operation per items. For example, a verifier can perform batch verification on n certificates faster than performing n verifications on a single membership proof. Aggregation is a batching technique that is used when non-interactively combining n items to a single item. For example, a prover can aggregate n membership proofs to a single constant size proof.

3 Security Issues of the Industrial Internet of Things

The security and privacy of the industrial Internet of Things is the focus of the Industrial Internet of Things security research. However, because these Industrial Internet of Things lack mutual trust mechanisms between devices, all devices are required to be checked against the data of the Industrial Internet of Things Center [37–39]. Once the database collapses, it will cause great damage to the entire Industrial Internet of Things and will there are a lot of data collection and transmission processes between sensor nodes in the network. Therefore, such systems often encounter security threats such as information leakage, information forgery and unauthorized access. The structure of the Industrial Internet of Things is generally divided into four layers, namely perception layer, on-site management layer, network layer and application layer [40]. The following is an analysis of data security issues associated with each layer.

3.1 Perception Layer Security Analysis

Sensing nodes are vulnerable to eavesdropping or control. The Industrial IoT sensing nodes are simple in function, low in processing power, and low in energy, unable to achieve complete security protection on their own, and the large number of node groups is not easy to manage and control, and is prone to omissions, which can be used by attackers machine [41–43]. Therefore, the communication information of the node is easy to be eavesdropped, and even the node may be controlled, so that the wrong information is sent and the network information is confused. In addition, if the gateway node is eavesdropped or controlled, it will directly cause the network to be paralyzed, and the entire network information will be leaked.

Node camouflage: Due to the fragility of the node and the variability of the network topology, an attacker could analyze a node to obtain its identity and password information, tampers with the hardware and software, and then captures the node, disguised as a legitimate user, can conduct illegal behavior or malicious attacks [44]; These include the monitoring of user information, replacing devices, publishing false information, launching DoS attacks, etc.

3.2 On-Site Management Security Analysis

The on-site management mainly refers to the local dispatch management center of the factory, which belongs to the centralized management mode. There is the possibility of data loss due to the failure of individual equipment [45]. When the amount of data information is too large, it will increase the burden on the central server. Industrial IoT data streams are aggregated into a single central control system. With the continuous evolution of low-power wide-area technology (LPWA), industrial IoT devices will grow geometrically in the future, and the cost of centralized services will be difficult to bear. Moreover, lack of security, it is easy to accept false information and make wrong decisions.

3.3 Network Layer Security Analysis

Due to the small amount of data transmitted by devices in the Industrial Internet of Things, complex encryption algorithms are generally not used to protect data. As a result, data is stolen, tampered, attacked, illegally accessed to the network during transmission, eavesdropping on the data, and destroying confidentiality and integrity; denial of service attacks, man-in-the-middle attacks, virus intrusion, use of sniffer tools and system vulnerabilities attacks and other attack methods [46–48].

3.4 Application Layer Security Analysis

The industrial IoT application layer stores a large amount of user data. How to effectively store data to avoid data loss or damage, how to isolate data from multi-tenant applications, how to avoid data services from being blocked, and how to quickly recover data after a failure are all security issues that need to be considered by the application layer.

Data access rights, user authentication: The application layer is a direct layer for interworking with users and provides users with data access rights. Therefore, a sound authentication mechanism and access permission settings to isolate the intrusion of illegal users is the key security point of the IoT application system [49].

User privacy leakage: Privacy issues represent the biggest obstacle in the implementation of the Internet of Things. The Internet of Things is involved with many aspects of a user's life. Once information is leaked, the user's property, information security, and personal privacy can easily be violated [50–52]. Ensuring the privacy of information is the primary issue to be solved to promote the development of the Industrial Internet of Things.

4 Secure Storage of Industrial IoT Data Based on Blockchain

4.1 Industrial Internet of Things Data Storage Based on Blockchain

Firstly, this paper deploys the blockchain in the factory to illustrate the data storage mechanism of the blockchain system, which is illustrated in Fig. 4 below.

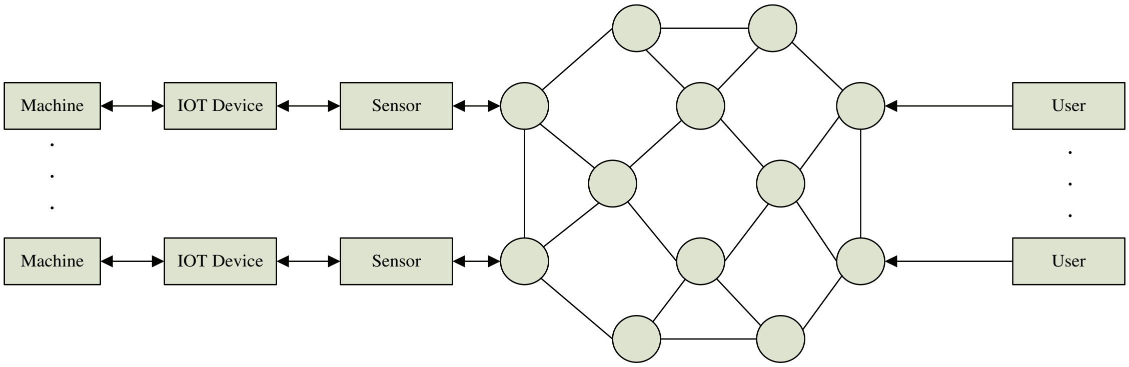

Figure 4: Data secure storage of industrial IoT based on blockchain

In the industrial IoT smart factory, various types of machines and users form the light nodes of the blockchain system. The management departments of various types of machines make up the full node of the blockchain system. A light chain node and a full node constitute a blockchain system.

The light node only stores the current state data in the blockchain system (the data collected at a certain moment, when the new data is collected, the data saved by the light node also changes). However, light nodes do not participate in the consensus and can transmit and receive data. In the blockchain system, full nodes not only save all data, but also participate in consensus. In addition, smart contracts are also deployed in the blockchain system to associate user information with management departments and machines to improve efficiency.

Data generated by various types of machines are transmitted to P2P networks through IoT devices and sensors. The data is continuously generated. At this time, the light node itself will store the current latest data and transmit the data to the full node. Each full node will verify the uploaded data and collect the data after verification. Then all the full nodes will reach a consensus and reach an agreement to store the data securely. The full node can also control how machine data is shared. User information involves two types of inquiries and orders. When users query information, because smart contracts are deployed in the blockchain network, the corresponding information can be obtained by triggering smart contracts to ensure that sensitive information is not leaked and access control is achieved. When the user has an order task, the user can unlock the smart contract by paying a deposit to the smart contract, and then send the corresponding information to each machine to complete the order task. This system can ensure that all uploaded data is true, stored data is not tampered with, decentralized, and automatically complete user data query and order tasks by deploying smart contracts.

4.2 Blockchain Data Storage Mechanism Based on Accumulator

Since the data in the traditional blockchain is stored using the Merkle hash tree, meaning that provide non-member proof cannot be provided. By contrast, the accumulator has the function of providing non-member proof. Accordingly, this paper proposes to use the accumulator instead of the original Merkel tree in the block to build an accumulator-based blockchain data storage mechanism.

In the improved blockchain in this paper, all full nodes are connected to an accumulator, and full nodes are used for data storage and data verification. The Merkle tree of each block is replaced with an accumulator. All light nodes are not verified, only the current state and data transmission are stored, and the Merkle tree is replaced with an accumulator.

Each block contains a block header and a block body. In addition to replacing the Merkle tree in the block body with an accumulator, this paper also allows each full node in the blockchain network to share an accumulator, but the light node does not. The accumulated value of the accumulator shared by all nodes is the value of all data in the entire blockchain, that is, assuming that the current block is the nth block, the accumulated value is $acc(X)$, and a new block is added at this time. The accumulator adds the data of the $n + 1$th block, and then the accumulated value $acc(X)$ also needs to be changed accordingly. The accumulated value of each block is the accumulated value of the data collected by this block in a certain period of time. The Merkle root hash value in the block header becomes the accumulated value, and an accumulated value 1 is added. This accumulated value is the accumulated value of the accumulator replacing the Merkle tree, and the Merkle tree in the block body becomes the accumulator. The data points $n_1, n_2, \ldots, n_m$ represent the data collected by the nodes over a certain period. At this time, the hash value is not computed in the block body, but the accumulated value is computed, and the obtained accumulated value exists in the block header. The advantage of this approach is that it can reduce storage memory. When the Merkle tree was originally used, the hash value of each layer was stored, and now an accumulated value is stored. In addition, the accumulator can also provide non-member proof.

The blockchain is jointly maintained by many network nodes. The improved blockchain proposed in this paper comprises an accumulator for each block, while each full node shares an accumulator, and the light node also has an accumulator, but the light node does not share an accumulator. Both accumulated values are stored in the block header, and each block connects with other block by finding the hash value of the block header.

First, the block data is created: A machine generates a set of data, along with a signature unique to the data uploading machine, and then uploads the data to the blockchain network, and other machines do the same. In other words, the data is generated continuously. At this time, all nodes of the blockchain network collect the data uploaded over a period of time to verify the legitimacy of the data source, pass the verification, and arrange the data in a certain order. Each light node only stores the current latest data state.

In the next step, a new block is created: After all the full nodes of the blockchain network have collected the data, a new block is created. At this time, the accumulator in the block generates the accumulated value of the data in the block, and then the accumulator also makes corresponding changes to the newly added data accumulation value, stores both accumulation values in the block header, after which the block header information is combined into a string. A 250 binary number is obtained through the hash functions twice, and the result is then generated. The difficulty value setting is met, if the first few digits are 0, if it is not satisfied, it must recalculate by adjusting the nonce value until it meets. Which is:

$$\text{Hblock header} \leq \text{target} \tag{1}$$

Once a certain full node in the blockchain network is calculated and the new block is created successfully, then this node will broadcast the successful block message to the entire network and other full nodes will receive the message.

The second element is node verification: When the full node receives the message released by the new block, the full node will verify it. At this time, the verification content comprises two parts: whether the data in the block is included in the accumulated value of the full node, and whether the hash value of the block header is less than the target difficulty. When all the full nodes are verified, a consensus is reached, and all other full nodes on the network agree to this newly generated block.

Finally, after the node verification is complete, the hash value obtained by the block header is connected to the hash value of the previous block header, after which the new block is successfully added to the blockchain.

In this paper, an accumulator is used to replace the Merkle tree in the block and each full node shares an accumulator, which can greatly reduce storage memory and is convenient for member and non-member verification. For example, the RSA accumulator is now used to prove the membership and non-membership, and the member certificate can be quickly provided directly through the accumulator, that is, $W_{x_i} = g^{\prod_{x_j \in X: x_j \neq x_i} r(x_j)} \ mod\ N$. For example, to prove that x_1 is in x_1, x_2, x_3, use the formula $W_{x_1} = g^{r(x_2)r(x_3)} \ mod\ N$, and find that x_1 is a member. When proving non-membership, it is assumed that x_1 is in (x_2, x_3, x_4), and the accumulative value $acc(X) = g^{r(x_2)r(x_3)r(x_4)}$. Calculate $ax_1 + bx_2x_3x_4 = 1$, find a and b. Verification check $g^{ar(x_1)} acc(X)^b = g^1$, prove that x_1 is a non-member factor. Instead of applying for the first node like Merkle tree, calculate the various hash values involved from the leaf node. If it is a light node verification, it will be necessary to apply to other nodes for additional hash values involved. The process is cumbersome.

In the improved blockchain in this paper, each full node is both a storage node and a verification node, and the light node only stores the current state. When you want to query historical data, you can query it through the accumulator of the full node, and you can also verify whether the data belongs to the originally collected data. If you want to query the specific historical data in a certain period of time, you can query the corresponding data through the block accumulator, whether it is the overall data query or the block data query.

4.3 Improved RSA Accumulator Definition

Because the accumulator generally has trapdoors and a single update element, it is difficult to meet the requirements of security and large-scale data addition. Accordingly, this paper improves the accumulator. An improved RSA accumulator is proposed so that the accumulator has no trapdoors and can update elements in batches. For newly added data, use batch to add to the accumulator. For elements to be deleted, use bulk deletes to remove useless data from the accumulator. Let k be a security parameter, and a batch update accumulator with no trapdoor consists of the following algorithms:

- *Setup* $(1^k, h)$: It is setting algorithm that security parameter k and random type h as input and randomly generates t.
- *KeyGen* $\left(1^k, t\right)$: It is a probabilistic algorithm that takes security parameters k and t as input and returns the parameter pk, where pk is the public key.
- $AccVal(X, PK)$: It is a probabilistic algorithm for computing accumulated values. It takes the set $X = \{x_1, x_2, \ldots, x_n\}$ and pk as input, and returns the accumulative value $acc(X)$ and auxiliary information a_c that some other algorithms will use.
- *Verify*$(x, W, acc(X), pk)$: It is a deterministic algorithm that uses the witness W and pk to check whether the element x belongs to the set X represented by the accumulated value $acc(X)$. Witness x of W effectively returns *Yes*, otherwise returns *No*.
- $AddEle(X^{\oplus}, a_c, acc(X), pk)$: It is a probabilistic algorithm that adds some new elements in batches. The input value is to add $X^{\oplus} = \left\{x_1^{\oplus}, x_2^{\oplus}, \ldots, x_l^{\oplus}\right\}$ element set, auxiliary information a_c, accumulative value $acc(X)$ and parameter pk. The return value is the accumulative value of $acc(X \cup X^{\oplus})$ and the set $X \cup X^{\oplus}$, the witness $\left\{W_1^{\oplus}, W_2^{\oplus}, \ldots, W_l^{\oplus}\right\}$ and the inserted element $\left\{x_1^{\oplus}, x_2^{\oplus}, \ldots, x_l^{\oplus}\right\}$ and auxiliary information a_c, a_u, which will be used for future update operations.

- *DelEle*$(X^{\ominus}, a_c, acc(X), pk)$: It is a probabilistic algorithm that to delete some elements in batches. The input value is to delete the element set of $X^{\ominus} = \{x_1^{\ominus}, x_2^{\ominus}, \ldots, x_l^{\ominus}\}$, auxiliary information a_c, accumulative value $acc(X)$ and parameter pk. The return value is the accumulative value of $acc(X \backslash X^{\ominus})$ corresponding to the set $X \backslash X^{\ominus}$, and the auxiliary information a_c, and a_u will be used for future update operation.
- *WitGen*(a_c, X, pk): It is a probabilistic algorithm that creates a witness for each element in set X, taking auxiliary information a_c, set X and parameter pk as inputs.
- *UpdWit*(W_i, a_u, pk): It is a deterministic algorithm that updates witnesses for $acc(X)$ and $acc(x')$ (new set after update) that are still accumulating. The input is W_i, the witness to be updated, auxiliary information a_c and pk. An updated witness $W_i^{'}$ is returned, and the witness can prove that x_i is still accumulating in the new accumulative value $acc(X')$.

4.4 Improved Accumulator Based on RSA

The non-trapdoor batch update accumulator proposed in this paper is implemented by means of RSA. The program contains eight parts: $\{$*Setup*, *KeyGen*, *AccVal*, *WitGen*, *Verify*, *AddEle*, *DelEle*, *UpdWit*$\}$. The scheme eliminates the trusted setting through Setup, introduces the representation of prime numbers, and can map common elements to prime numbers, and the scheme can also be updated in batches. The specific scheme is described in Section 4.4.4.

4.4.1 Specific Scheme

- *Setup* $(1^k, h)$: It is setting algorithm, and this paper uses no feasible setting. Generate *t* randomly by *h.*
- *KeyGen* $\left(1^k, t\right)$: Given a security parameter *k* and a random type *t*, an appropriate security modulus *N* is generated, $N = pq$, where *p*, *q* are strong prime numbers, and $g \in QR_N$ is used for exponentiation. Given two general hash functions $h(x)$, which are used to compute the prime representation, the function $h(x)$ maps the *k*-bit element to a 3*k*-bit element. Finally, a *pk* is returned.
- *AccVal*(X, pk): Given an element set $X = \{x_1, x_2, \ldots, x_n\}$ and *pk* as input. Use $h(x)$ to make the set $X = \{x_1, x_2, \ldots, x_n\}$ into a3*k*-bit element $\{r(x_1), r(x_2), \ldots, r(x_n)\}$, compute: $acc(X) = g^{r(x_1)r(x_2)\ldots r(x_n)}\ mod\ N$, The accumulated value is obtained, and finally the accumulated value $acc(X)$ and auxiliary information a_c are output.
- *WitGen*(a_c, X, pk): Given a_c, X and *pk*. Use $h(x)$ to make the set $X = \{x_1, x_2, \ldots, x_n\}$ into a 3*k*-bit element $\{r(x_1), r(x_2), \ldots, r(x_n)\}$, compute: $W_{x_i} = g^{\prod_{x_j \in X : x_j \neq x_i} r(x_j)}\ mod\ N$, Witness (W_i, x_i) of each element is output.
- *Verify*$(x, W, acc(X), pk)$: Given an element *x*, its witness *W*, the accumulated value $acc(X)$ and *pk*, check whether the element *x* belongs to the set *X* represented by the accumulated value $acc(X)$ and $W^x = acc(X)$. If yes, return *Yes*, otherwise return *No*.
- *AddEle*$(X^{\oplus}, a_c, acc(X), pk)$: Given a set of elements $X^{\oplus} = \{x_1^{\oplus}, x_2^{\oplus}, \ldots, x_l^{\oplus}\}$ as insert, auxiliary information a_c, accumulative value $acc(X)$ and *pk*, compute: $acc(X') = acc(X)^{r\left(x_1^{\oplus}\right)r\left(x_2^{\oplus}\right)\ldots r\left(x_l^{\oplus}\right)}\ mod\ N$ $W_i^{\oplus} = W_{x_i}^{x_i} g^{\prod_{x_i^{\oplus} \in X \cup X^{\oplus} : x_i^{\oplus} \neq x_j^{\oplus}} r\left(x_i^{\oplus}\right)}\ mod\ N$. Then output a new accumulated value $acc(X')$, witness $(W_i^{\oplus}, x_i^{\oplus})$, and auxiliary information a_c, a_u.
- *DelEle*$(X^{\ominus}, a_c, acc(X), pk)$: Given a set of elements $X^{\ominus} = \{x_1^{\ominus}, x_2^{\ominus}, \ldots, x_l^{\ominus}\}$ as deletion, auxiliary information a_c, accumulative value $acc(X)$ and *pk*, compute: $acc(x') = acc(X)^{\backslash r\left(x_1^{\oplus}\right)r\left(x_2^{\oplus}\right)\ldots r\left(x_l^{\oplus}\right)}\ mod\ N$. Then output a new accumulated value $acc(x')$ corresponding to the set $X \backslash X^{\ominus}$ and auxiliary information a_c, a_u.

- *UpdWit*(W_i, a_u, pk): Given a witness W_i, auxiliary information a_c and pk, compute $W_{x_i} = g^{\prod_{x_j \in X : x_j \neq x_i} r(x_j)}$ *mod N*, then output a new one for each element x_i witness $W_i^{'}$.

4.4.2 Correctness

The correct property of the accumulator scheme is just to say that if the element x belongs to the accumulation set X, and if the corresponding witnesses W have been computed using *WitGen* and *UpdWit*, the verification process should pass. The scheme {*Setup*, *KeyGen*, *AccVal*, *WitGen*, *Verify*, *AddEle*, *DelEle*, *UpdWit*} is correct. We say that for all sufficiently large $k \in N$, for the *pk* output by Algorithm *Setup*() and Algorithm *KeyGen*(), both *AccVal* and *WitGen* algorithms will output the correct accumulated value and witness, and the *AddEle* and *DelEle* algorithms will output the new accumulative values and witnesses, all probabilities in k can be negligible.

Proof: First, we show that the Verify algorithm is correct for the accumulator. Let set $X = \{x_1, x_2, \ldots, x_2\}$, $acc(X)$ is the corresponding accumulated value, and *pk* is the public key. Where x is considered to be the i-th element of the set, because:

$$acc(X) = g^{r(x_1)r(x_2)\ldots r(x_n)} \; mod\ N \tag{2}$$

$$W_{x_i} = g^{\prod_{x_j \in X : x_j \neq x_i} r(x_j)} \; mod\ N \tag{3}$$

so

$$W^x = W_{x_i}^{x_i} = g^{\prod_{x_j \in X : x_j \neq x_i} r(x_j) r(x_i)} \; mod\ N = g^{r(x_1)r(x_2)\ldots r(x_n)} \; mod\ N = \; acc(X) \tag{4}$$

therefore

$$W^x = acc(X) \tag{5}$$

Consistency shows that in the accumulator, if the element x is accumulated in the accumulated value, the witness can provide a valid proof for x.

When some elements are added, for the newly added element $x_i^{\oplus} (i = 1, \ldots, l)$, it is easy to verify the correctness in the same way; for the old element x_i whose witness W_i is updated to $W_i^{'}$, the correctness is as follows:

$$W_i^{'x_i} = W_{x_i}^{x_i} W_i^{\oplus} = \left(g^{\prod_{x_j \in X : x_j \neq x_i} r(x_j) r(x_i)} \; mod\ N \right) (g^{\prod_{x_i^{\oplus} \in X \cup X^{\oplus} : x_i^{\oplus} \neq x_j^{\oplus}} r\left(x_i^{\oplus}\right)} \; mod\ N$$

$$= g^{r(x_1)r(x_2)\ldots r(x_n)} g^{\prod_{x_i^{\oplus} \in X \cup X^{\oplus} : x_i^{\oplus} \neq x_j^{\oplus}} r\left(x_i^{\oplus}\right)} \; mod\ N = acc(X)^{r\left(x_1^{\oplus}\right) r\left(x_2^{\oplus}\right) \ldots r\left(x_l^{\oplus}\right)} \; mod\ N = \; acc(X') \tag{6}$$

For deleting elements, you can verify the correctness in the same way.

4.4.3 Security

The security of the accumulator scheme is illustrated by an experiment in which the adversary plays the role of user and attempts to forge witnesses (i.e., find valid witnesses for elements that do not belong to the set). Such opponents must succeed with a very low probability. If the opponent finds a set of elements $X = \{x_1, x_2, \ldots, x_n\} \subseteq S$, where S is the domain, the element $x' \in S \backslash X$ and a witness W' can prove that x' has accumulated in the accumulated value The possibility is negligible.

The security of the accumulator is based on strong *RSA*. Given an *RSA* modulus N and a random element $x \in Z_N$, this is difficult (i.e., it can be done with $neg(k)$, which can be negligible in the security parameter k) Say, it can find $y > 1$, so $a^y = x \; mod\ N$.

Let k be the security parameter, h be the double universal hash function, and N be the $(3w + 1)$ bit RSA module. Given a set of elements X, a computationally-bounded opponent A, can find a set X' with the same accumulated value as X(i.e., $acc(X') = acc(X)$) is negligible.

Proof: Suppose A can find a set X', which means that A finds other sets $\{x_1^{'}, x_2^{'}, \ldots, x_{n'}^{'}\} \neq \{x_1, x_2, \ldots, x_n\}$ such that $g^{r(x_1)r(x_2)\ldots r(x_n)} = g^{r\left(x_1^{'}\right)r\left(x_2^{'}\right)\ldots r\left(x_{n'}^{'}\right)}\ mod\ N$. By constructing a prime number representative, it is impossible to associate the lead representative with two different elements, therefore, it also keeps $\{r(x_1), r(x_2), \ldots, r(x_n)\} \neq \{r\left(x_1^{'}\right), r\left(x_2^{'}\right), \ldots, r\left(x_{n'}^{'}\right)$, which means that the opponent can find a value A and an index j such that:

$$A^{r\left(x_j\right)} = g^{r\left(x_1^{'}\right)r\left(x_2^{'}\right)\ldots r\left(x_{n'}^{'}\right)}\ mod\ N \tag{7}$$

where: $A = g^{\prod_{i \neq j} r(x_i)}\ mod\ N$. Now let $x = r\left(x_j\right)$ and $r = r\left(x_1^{'}\right)r\left(x_2^{'}\right) \ldots r\left(x_{n'}^{'}\right)$, the opponent can now compute the x-th root of g as follows, Since $r\left(x_j\right)$ is a prime number, A uses the extended euclidean algorithm to compute $a, b \in Z$ such that $ar + br\left(x_j\right) = 1$. Now let $y = A^a g^b, y^x = A^{ar\left(x_j\right)} g^{br\left(x_j\right)} = g^{ar+br\left(x_j\right)} = g\ mod\ N$. Therefore, A can break the strong RSA hypothesis that appears with probability $v(k)$ bit $neg(k)$ is negligible.

Security without trusted settings: Let $\{Setup$, $KeyGen$, $AccVal$, $WitGen$, $Verify$, $Verify$, $AddEle$, $DelEle$, $UpdWit\}$ be accumulators. If the following conditions are met, the accumulator has no trusted settings:

$$\Pr\left[t \leftarrow Setup\left(1^k, h\right), pk \leftarrow Gen\left(1^k, t\right), (x, X, W) \leftarrow A(h, t, pk) : x \notin X \wedge verify(x, W) = member\right]$$

$$= neg(k) \tag{8}$$

for any PPT opponent A. The accumulator is no trusted setting, if

$$\begin{aligned}&\Pr[t \leftarrow Setup(1^k, h), pk \leftarrow Gen(1^k, t), (x, X, W) \leftarrow A(h, t, pk) :\\&\quad (verify(x, W) = member) \wedge (verify(x, W') = nonmember)] = neg(k)\end{aligned} \tag{9}$$

for any PPT opponent A.

Note that in this model, t is chosen honestly, while the opponent can choose t and h by themselves. In fact, a more stringent requirement can be considered, where h is not only known by the opponent, but actually chosen by her. However, in this case, in all subsequent security assumptions, even if the adversary can choose the basic module, they must also assume that these assumptions hold. Unfortunately, no module family has been established under this security assumption.

5 Conclusion

This paper first analyzes the challenges of industrial IoT data security, then analyzes the blockchain technology, and proposes a blockchain-based industrial IoT data security storage mechanism. The data layer of the blockchain uses a Merkel tree to store data, but the Merkel tree cannot provide proof of non-membership. Accumulate the advantages of strength, universality and compactness of appliances. It can provide non-member certification, reduce data storage overhead, and better protect privacy. Therefore, this paper builds an accumulator-based blockchain data storage. But the traditional accumulator has trapdoors and cannot batch update elements. Accordingly, to solve the problem of data storage expansion in the blockchain, the concept of an improved RSA accumulator is proposed. It not only has no trapdoor, but also can add and delete elements in batches. Finally, the validity of the proposed scheme is proved by correctness and security.

Funding Statement: This work is supported by the NSFC (61772454, 61772280, 62072249), and by the National Key Research and Development Program of China (2019YFC1511000).

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] K. R. Choo, S. Gritzalis and J. H. Park, "Cryptographic solutions for Industrial Internet-of-Things: Research challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, 2018.

[2] W. P. Nwadiugwu and D. Kim, "Energy-efficient sensors in data centers for Industrial Internet of Things (IIoT)," in *International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, *Bhimtal*, pp. 1–6, 2018.

[3] X. Zhang, J. Duan, W. Sun and S. Jha, "A tumour perception system based on a multi-layer mass-spring model," *International Journal of Sensor Networks*, vol. 31, no. 1, pp. 24–32, 2019.

[4] D. Cao, Y. C. Jiang, J. Wang, B. F. Ji, O. Alfarraj *et al.,* "Adaptive relay-node selection method for message broadcasting in the internet of vehicles," *Sensors*, vol. 20, no. 5, pp. 1338, 2020.

[5] J. Wang, Y. N. Tang, S. M. He, C. Q. Zhao, P. K. Sharma *et al.,* "LogEvent2vec: LogEvent-to-Vector based anomaly detection for large-scale logs in Internet of Things," *Sensors*, vol. 20, pp. 2451, 2020.

[6] D. Cao, B. Zheng, B. F. Ji, Z. B. Lei and C. F. Feng, "A robust distance-based relay selection for message dissemination in vehicular network," *Wireless Networks*, vol. 26, no. 3, pp. 1755–1771, 2020.

[7] S. M. H. Rostami, A. K. Sangaiah, J. Wang and X. Z. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *Eurasip Journal on Wireless Communications & Networking*, vol. 70 , pp. 1–19, 2019.

[8] Y. Mao, J. Zhang, H. Qi and L. Wang, "DNN-MVL: DNN-multi-view-learning-based recover block missing data in a dam safety monitoring system," *Sensors*, vol. 19, no. 13, pp. 2895, 2019.

[9] T. Liu, G. Huang and P. Zhang, "A user authentication protocol combined with the trust model biometrics and ECC for wireless sensor networks," *Intelligent Automation and Soft Computing*, vol. 24, no. 3, pp. 519–529, 2018.

[10] E. A. Saksonov, Y. L. Leokhin and V. N. Azarov, "Organization of information security in Industrial Internet of Things systems," in *Int. Conf. Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*, *Sochi, Russia*, pp. 3–7, 2019.

[11] M. Z. Hasan and H. Al-Rizzo, "Optimization of sensor deployment for Industrial Internet of Things using a multiswarm algorithm," in *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10344–10362, 2019.

[12] L. M. Fang, Y. Li, X. Y. Yun, Z. Y. Wen, S. L. Ji *et al.,* "THP: A novel authentication scheme to prevent multiple attacks in SDN-based IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5745–5759, 2020.

[13] H. Hui, C. Zhou, S. Xu and F. Lin, "A novel secure data transmission scheme in industrial Internet of Things," *China Communications*, vol. 17, no. 1, pp. 73–88, 2020.

[14] W. Zhao, J. Liu and H. Guo, "Etc-iot: Edge-node-assisted transmitting for the cloud-centric Internet of Things," *IEEE Network*, vol. 32, no. 3, pp. 101–107, 2018.

[15] A. Karmakar, N. Dey, T. Baral, M. Chowdhury and M. Rehan, "Industrial Internet of Things: A review," in *Int. Conf. on Opto-Electronics and Applied Optics (Optronix)*, Kolkata, India, pp. 1–6, 2019.

[16] H. Chen, M. Hu, H. Yan and P. Yu, "Research on Industrial Internet of Things security architecture and protection strategy," in *Int. Conf. on Virtual Reality and Intelligent Systems*, Jishou, China, pp. 365–368, 2019.

[17] J. Robert, S. Rauh, H. Lieske and A. Heuberger, "IEEE 802.15 low power wide area network (LPWAN) PHY interference model," in *Proc. IEEE Int. Conf. Commun.*, Kansas City, MO, USA, pp. 1–6, 2018.

[18] Y. J. Ren, Y. Leng, F. J. Zhu, J. Wang and H. J. Kim, "Data storage mechanism based on blockchain with privacy protection in wireless body area network," *Sensors*, vol. 19, no. 10, 2395, 2019.

[19] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah and S. J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, pp. 2579, 2019.

[20] Y. S. Zhou, X. W. Long, L. J. Chen and Z. Yang, “Conditional privacy-preserving authentication and key agreement scheme for roaming services in VANETs,” *Journal of Information Security and Applications*, vol. 47, pp. 295–301, 2019.

[21] Y. J. Ren, F. J. Zhu, P. K. Sharma, T. Wang, J. Wang *et al.,* “Data query mechanism based on hash computing power of blockchain in Internet of Things,” *Sensors*, vol. 20, no. 1, 207, 2020.

[22] Y. J. Ren, Y. P. Liu, S. Ji, A. K. Sangaiah and J. Wang, “Incentive mechanism of data storage based on blockchain for wireless sensor networks,” *Mobile Information Systems*, vol. 2018, pp. 1–10, 2018.

[23] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad and J. Wang, “Blockchain enabled distributed security framework for next generation IoT: An edge-cloud and software defined network integrated approach,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6143–6149, 2020.

[24] B. Yin and X. T. Wei, “Communication-efficient data aggregation tree construction for complex queries in IoT applications,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3352–3363, 2018.

[25] Q. Wang and Y. G. Wang, “Research on power Internet of Things architecture for smart grid demand,” in *2018 2nd IEEE Conf. on Energy Internet and Energy System Integration (EI2)*, Beijing, pp. 1–9, 2018.

[26] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen and B. C. Ooi, “Untangling blockchain: A data processing view of blockchain systems,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[27] Z. Q. Xia, J. J. Tan, J. Wang, R. L. Zhu, H. G. Xiao *et al.,* “Research on fair trading mechanism of surplus power based on blockchain,” *Journal of Universal Computer Science*, vol. 25, no. 10, pp. 1240–1260, 2019.

[28] J. Y. Zhang, S. Q. Zhong, T. Wang, H. C. Chao and J. Wang, “Blockchain-based systems and applications: A survey,” *Journal of Internet Technology*, vol. 21, no. 1, pp. 1–14, 2020.

[29] K. Christidis and M. DevetsikIoTis, “Blockchains and smart contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[30] W. Wang, D. T. Hoang, P. Hu, Z. Xiong and D. Niyato, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[31] J. Liu and Z. Liu, “A survey on security verification of blockchain smart contracts,” *IEEE Access*, vol. 7, pp. 77894–77904, 2019.

[32] Y. J. Ren, Y. Leng, Y. P. Cheng and J. Wang, “Secure data storage based on blockchain and coding in edge computing,” *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 1874–1892, 2019.

[33] Y. Yu, Y. Li, J. Tian and J. Liu, “Blockchain-based solutions to security and privacy issues in the Internet of Things,” *IEEE Wireless Communications*, vol. 25, no. 6, pp. 12–18, 2018.

[34] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer and M. Virza, “SNARKs for C: Verifying program executionssuccinctly and in zeroknowledge,” in *Proceedings of International Cryptology Conf.*, Springer, Lecture Notes in Computer Science, vol. 8043, pp. 90–108, 2013.

[35] M. T. Goodrich, R. Tamassia and J. Hasic, “An efficient dynamic and distributed cryptographic accumulator,” in *Proceedings of Information Security Conf.*, Springer, Lecture Notes in Computer Science, vol. 2433, pp. 372–388, 2002.

[36] J. Li, N. Li and R. Xue, “Universal accumulators with efficient nonmembership proofs,” *Proceedings of Applied Cryptography and Network Security*, vol. 4521, pp. 253–269, 2007.

[37] F. Y. Li, R. Xie, Z. Y. Wang, L. L. Guo, J. Ye *et al.,* “Online distributed IoT security monitoring with multidimensional streaming big data,” in *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4387–4394, 2020.

[38] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt and L. Wang, “Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs,” *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.

[39] J. Xu, Y. J. Zhang, K. Y. Fu and S. Peng, “SGX-based secure indexing system,” *IEEE Access*, vol. 7, pp. 77923–77931, 2019.

[40] Y. Jiang, M. H. Zhao, C. Q. Hu, L. L. He, H. T. Bai *et al.,* “A parallel Fp-growth algorithm mining world ocean atlas data using multi-core CPU,” *Journal of Supercomputing*, vol. 75, no. 2, pp. 732–745, 2019.

[41] J. Wang, W. B. Wu, Z. F. Liao and L. Wang, “An energy-efficient off-loading scheme for low latency in collaborative edge computing,” *IEEE Access*, vol. 7, pp. 149182–149190, 2019.

[42] J. Wang, X. J. Gu, W. Liu, A. K. Sangaiah and H. J. Kim, “An empower hamilton loop-based data collection algorithm with mobile agent for WSNs,” *Human-Centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–14, 2019.

[43] X. F. Wang, L. Wang, Y. H. Zheng and J. Wang, “An event-driven plan recognition algorithm based on intuitionistic fuzzy theory,” *Journal of Supercomputing*, vol. 74, no. 12, pp. 6923–6938, 2018.

[44] C. X. Wang, X. Shao, Z. Gao, C. X. Zhao and J. Gao, “Common network coding condition and traffic matching supported network coding aware routing for wireless multihop network,” *International Journal of Distributed Sensor Networks*, vol. 15, pp. 1–20, 2019.

[45] Y. Liu, R. Candell, M. Kashef and L. Benmohamed, “Dimensioning wireless use cases in Industrial Internet of Things,” in *2018 14th IEEE Int. Workshop on Factory Communication Systems*, Imperia, pp. 1–4, 2018, 2018,

[46] J. Wang, W. B. Wu, Z. F. Liao, R. S. Sherratt, G. J. Kim *et al.,* “A probability preferred priori offloading mechanism in mobile edge computing,” *IEEE Access*, vol. 8, no. 1, pp. 39758–39767, 2020.

[47] J. Wang, Y. Q. Yang, T. Wang, R. S. Sherratt and J. Y. Zhang, “Big data service architecture: A survey,” *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[48] G. S. Li, Y. C. Liu, J. H. Wu, D. D. Lin and S. S. Zhao, “Methods of resource scheduling based on optimized fuzzy clustering in fog computing,” *Sensors*, vol. 19, no. 9, pp. 1–16, 2019.

[49] Q. Y. Zhou and J. J. Luo, “The study on evaluation method of urban network security in the big data era,” *Intelligent Automation and Soft Computing*, vol. 24, no. 1, pp. 133–138, 2018.

[50] Y. Zhou, T. Liu, F. Tang and M. Tinashe, “An unlinkable authentication scheme for distributed IoT application,” *IEEE Access*, vol. 7, pp. 14757–14766, 2019.

[51] W. N. Wan, J. Chen and S. B. Zhang, “A cluster correlation power analysis against double blinding exponentiation,” *Journal of Information Security and Applications*, vol. 48, no. 10, pp. 1–8, 2019.

[52] L. Gong, B. Yang, T. Xue, J. Chen and W. Wang, “Secure rational numbers equivalence test based on threshold cryptosystem with rational numbers,” *Information Sciences*, vol. 466, pp. 44–54, 2018.