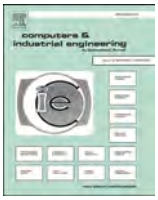
Computers & Industrial Engineering 164 (2022) 107903

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie



# Data security storage mechanism based on blockchain industrial Internet of Things

Jin Wang [a,b], Jiahao Chen [a], Yongjun Ren [c,*], Pradip Kumar Sharma [d], Osama Alfarraj [e], Amr Tolba [f]

[a] School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China
[b] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410004, China
[c] School of Computer Science, Engineering Research Center of Digital Forensics of Ministry of Education, Nanjing University of Information Science & Technology, Nanjing 210044, China
[d] Department of Computing Science, University of Aberdeen, UK
[e] Department of Computer Science, Community College, King Saud University, 11437, Riyadh, Saudi Arabia
[f] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia

ARTICLE INFO

*Keywords:*
Industry 5.0
Industrial Internet of Things
Blockchain
Data storage
Cryptographic commitment

ABSTRACT

In the age of Industry 5.0, the industrial Internet of Things (IIoT) system has changed from the original "cyber-physical" system to a complex "human-cyber-physical" system, data security issues become more important. Blockchain technology can be used to ensure the security of IIoT data. However, the traditional blockchain system uses Merkle trees to store data, in which the proof size is large when verifying the integrity and correctness of the data. And it is unable to perform batch verification of the data. Large size proof will bring great pressure to communication, causing end-to-end communication delays, which seriously affect the stability, efficiency, and security of IIoT system. To solve it, in the paper, the incremental aggregator subvector commitment (IASVC) is used to replace Merkle tree, which reduces the size of proof and communication consumption. Each block processes 1000 transactions, the proof size of a single data piece is 15% of the original scheme. Moreover, our scheme can realize the aggregation verification of the proof. In addition, the qualifications of data upload on nodes are set using IASVC, which can reduce the storage pressure of nodes by storing a single commitment instead of the entire qualification list.

## 1. Introduction

Industry 5.0 is considered to be the fifth industrial revolution, which will further enhance the integration of information and physical systems and the full integration of industry and human society (Demir, Döven, & Sezen, 2019; Özdemir, 2018; Wang, Gao, Yin, Li, & Kim, 2018). The biggest feature of Industrial 5.0 is "personalized customization". In the customer's personalized customization, production, inventory management and sales, the information systems of each subject are basically independent of each other. As a result, there will be problems with data fraud. If the equipment makes decisions based on the wrong information, it may bring serious economic and security problems. Second, traditional industrial IoT systems are based on centralized architecture, and the attack on the central node may lead to a large number of user data leakage or tampering. In addition, the number of sensor devices deployed in industrial 5.0 is much larger than that in industrial 4.0, and the centralized management mode will become unaffordable (Ren et al., 2020; Skobelev & Borovik, 2017). The combination of blockchain technology and IIoT technology is considered to be an effective way to solve the above problems (Alladi et al., 2019; Huang et al., 2019; Ren et al., 2018).

While blockchain provides a secure data storage environment for the IoT, it also brings a large number of data verification requirements for the IoT system. In order to ensure the availability of data, the blockchain-based IoT system needs to verify every piece of data to be used. The data layer of a traditional blockchain system uses a Merkle

* Correspondence author at: School of Computer Science, Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science & Technology, Nanjing 210044, China.
*E-mail addresses:* jinwang@csust.edu.cn (J. Wang), renyj100@126.com (Y. Ren), pradip.sharma@abdn.ac.uk (P.K. Sharma), oalfarraj@ksu.edu.sa (O. Alfarraj), atolba@ksu.edu.sa (A. Tolba).

https://doi.org/10.1016/j.cie.2021.107903

Available online 25 December 2021

tree to store data. Verify the availability of data with Merkle proofs when using the data. But Merkle's proof can only verify the availability of a single data piece (Kate et al., 2010). When verifying the availability of large amounts of data, a large number of proofs need to be generated. Tremendous number of proofs will bring great communication pressure to nodes, which may lead to communication delays. Communication delays will endanger the stability and security of the IoT system (Yan et al., 2012). In order to make the blockchain better used in the IoT system, it is necessary to reduce the overhead of verifying the availability of data. The vector commitment provides a new idea for reducing communication costs.

Vector commitment is a special kind of commitment. Batchable vector commitment has been used in keyless proofs of retrievability (PoR), concise argument based on probabilistic checkable proof (PCP), interactive oracle proof (IOP), verifiable decentralized storage (VDS) and other applications to reduce communication cost (Boneh, Bünz, & Fisch, 2019; Campanelli, Fiore, Greco, Kolonelos, & Nizzardo, 2020; Lai & Malavolta, 2019). Batchable vector commitment is considered as a useful tool to reduce communication consumption. To reduce communication overhead, Lai and Malavolta (2019) and Boneh et al. (2019) have used subvector commitment instead of Merkle tree to commit PCP strings. The main idea of this paper is to use IASVC to replace the Merkle tree in the blockchain to reduce the communication consumption. The reduction of communication consumption can reduce the communication pressure of nodes and improve the stability of the system. The concept of vector commitment first appeared in (Catalano et al., 2011; Kate et al., 2010; Libert & Yung, 2010). Vector commitment was formalized by Catalano and Fiore (2013). Catalano's scheme is based on computational Diffie-Hellmen (CDH) and RSA assumption. Like the Merkle tree, Catalano et al.'s scheme verifies only one data piece per proof. Boneh et al. (2019) used hidden-order groups to construct subvector commitment. Their scheme is the first scheme that allows multiple proofs to be aggregated under certain conditions, and it is also the first scheme with constant-sized public parameters. Their scheme can only achieve one-hop aggregation of the proof. Later, Lai and Malavolta (2019) expanded Catalano's scheme, formalized the subvector commitment, and realized the *I*-subvector proof with a constant proof size. But their scheme cannot achieve the disaggregation of proof. Campanelli et al. (2020) proposed the concept of IASVC, and constructed two kinds of IASVC based on RSA in the hidden order group. Their scheme achieves proof aggregation and disaggreation, which further reduces communication consumption. However, under the same security level, the single proof size of the hidden order group is much larger than that of the bilinear group scheme. Therefore, we construct the IASVC scheme in bilinear groups. Our scheme has a smaller proof size. In addition, Tomescu et al. (2020) used polynomial commitments and Lagrangian polynomials to construct a subvector commitment scheme, and used it in account-based stateless cryptocurrency to reduce the storage pressure of nodes on user balance states. Gorbunov et al. (2020) constructed a vector commitment with pointproof function and used it to reduce the storage consumption of smart contracts. The IASVC scheme in this paper can set the qualification of data uploading of sensor nodes in a similar way, thus reducing the storage pressure of nodes on the data uploading qualification list.

In order to understand the work of this paper more intuitively, we summarize the work of this paper in Fig. 1.

The contributions of this paper are summarized as follows.

- First, this paper analyzes the problems existing in the application of the traditional blockchain system in the IIoT and proposes to replace the Merkle tree in the blockchain data layer with IASVC. In this way, proofs of multiple data pieces can be aggregated to reduce the communication cost of verifying data availability.
- Second, to further reduce the proof size, we construct the IASVC scheme in the bilinear group. At the same level of security, the bilinear group scheme has proof of smaller size. In addition, our scheme can also be used to set node data upload qualification. Our scheme replaces storing the entire qualification list by storing a single commitment, which reduces the storage pressure on the qualification list.
- Thirdly, we analyzed the correctness of the aggregation and disaggregation algorithm, established the security model for our scheme and conducted a provable security analysis. By reducing the

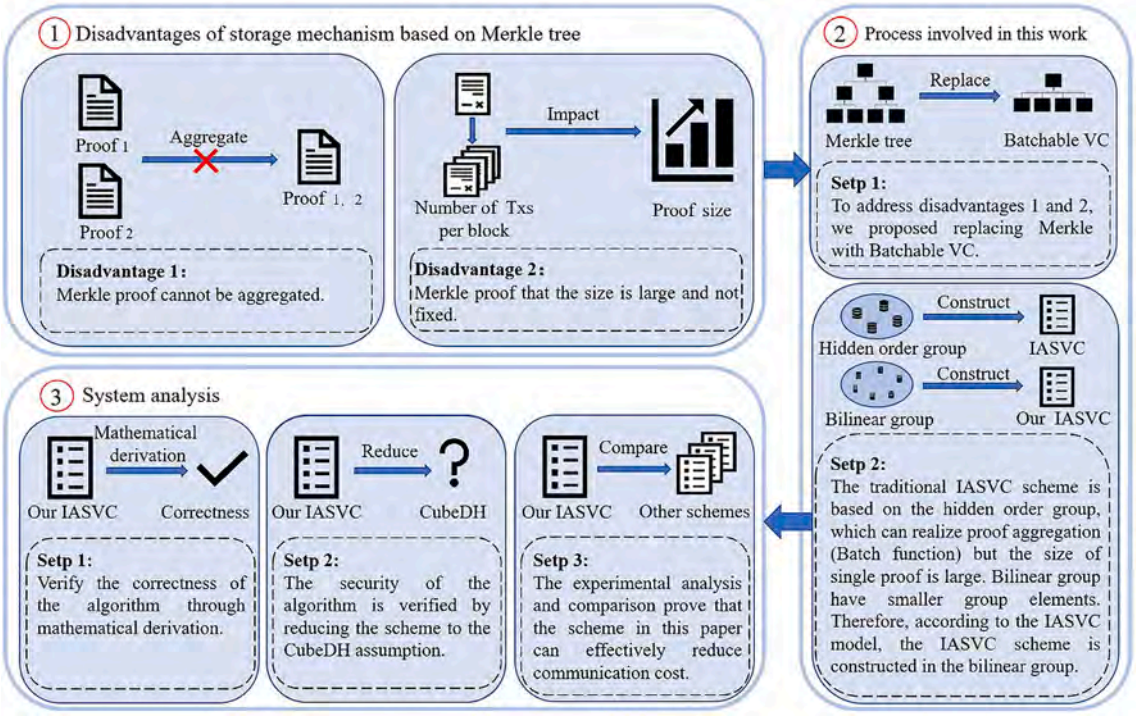



**Fig. 1.** Overview of work in this paper.

proposed scheme to CubeDH assumption, the security of the proposed scheme is proved.
- Finally, the performance of the proposed scheme is analyzed experimentally, and our scheme is compared with other schemes under 128-bit security. Experimental results show that our scheme can effectively reduce communication consumption when verifying data availability.

The rest of this paper is organized as follows. Section 2 introduces the knowledge of application of integration of blockchain and IIoT in Industry 5.0, and vector commitment. Section 3 presents the advantages and challenges of blockchain-enabled IIoT. Section 4 will introduce our proposed data storage mechanism. Section 5 will compare and analyze this scheme and other schemes. Finally, we conclude in Section 6.

## 2. Related work

In this section, we introduce the background knowledge related to the research in this paper. In Section 2.1, we will introduce IIoT and blockchain, and then describe the blockchain-enhanced IIoT architecture in detail. In Section 2.2, we reviewed the knowledge related to vector commitment.

### 2.1. Integration of blockchain and IIoT in Industry 5.0

In the age of Industrial 5.0, human, things, and computing devices will all be connected through the IIoT. Human will work with collaborative robots, which are responsible for trivial, repetitive and dangerous work, while human is responsible for product design and processes. In addition, customers can communicate with the factory in real time so that they can customize personalized products according to their individual needs (Faruqi, 2019; Nahavandi, 2019; Wang et al., 2019; Zhou et al., 2019). Industry 5.0 brings "human" into IIoT systems, which has led to a huge increase in the number of sensors connected to IoT systems. These large amounts of sensors will bring huge operating pressure and management costs. This is a huge challenge for the traditional IIoT system based on central architecture. Moreover, the addition of the element of "human" also makes the IIoT have to face privacy and security issues. The emergence of blockchain technology provides a new means to solve the appeal problem (Raikwar, Gligoroski, & Kralevska, 2019; Wang, Yang, Wang, Sherratt, & Zhang, 2020).

Blockchain is a special database technology that uses distributed storage methods, combined with point-to-point transmission, consensus mechanism, cryptographic algorithms and other technologies. Blockchain technology realizes the establishment of a distributed data ledger in an untrusted environment (Nofer, Gomber, Hinz, & Schiereck, 2017; Ren, Leng, Cheng, & Wang, 2019; Wang, Gao, Zhou, Simon Sherratt, & Wang, 2020). The characteristics of decentralized and weakly centralized blockchain can effectively reduce the operating pressure of central nodes. The immutable and traceability features of blockchain enable it to guarantee the secure storage of data. Many scholars have used blockchain to improve various IoT applications (Pieroni et al., 2018; Ren et al, 2021). In drug production, smart drug production can be achieved by using blockchain-based IoT systems. A wearable device tracks vital signs in real time and provides feedback. This provides more secure and more accurate data for drug production (Gong et al, 2018, Salahuddin et al., 2018). In terms of supply chain, the IIoT is applied to raw material procurement, inventory management, personalized customized sales and other aspects. The use of blockchain can provide information traceability for various information subjects (Mondal et al., 2019). In smart factory, blockchain can be used for data interaction in the production process to ensure the availability of data for use (Sharma et al., 2018). In the food industry and product management, blockchain is used for product traceability. In the event of food security problems, blockchain is used to accurately track the responsible parties (Fernández-Caramés et al., 2018; Ge et al., 2019; Lin et al., 2019).

Fig. 2 shows the architecture after blockchain is integrated into IIoT (Boyes, Hallaq, Cunningham, & Watson, 2018; Ren, Qi, Cheng, Wang, & Alfarraj, 2020). The perception layer is mainly responsible for the perception of industrial machine information, user parameters and the release of control instructions. The perceptual layer consists of various data collection devices. Data collection equipment mainly refers to all kinds of intelligent sensors with communication function, RFID, operation equipment, etc. (Khattak et al., 2019; Xu et al., 2019; Wang et al., 2018). The communication layer is the transport layer of the information of the IIoT. The information transmission technology of the communication layer includes 5G, low power wide area network, ZigBee network and so on. The application layer is the data processing center. According to the demand of industrial application, big data analysis is used to realize the functions of collaborative manufacturing, mass customization, production optimization, inventory management and so on (Aslam et al., 2020; Mao et al., 2019; Ren et al., 2019; Wang et al., 2016).

The blockchain composite layer includes five sub-layers. The blockchain data layer is responsible for data collection and preprocessing. The data collected by the perception layer will be pre-processed according to application requirements. Preprocessing includes Hash operation, asymmetric encryption, digital signature, etc. Then send the processed data to the network layer for propagation. The blockchain network layer is essentially a distributed network running on the communication layer. This layer is mainly composed of propagation protocols, overlay routing and verification mechanisms. The incentive layer is not a necessary element in the architecture. This layer is responsible for providing rewards to nodes that contribute to the consensus process. The blockchain service layer provides docking services between IIoT applications and blockchain components. Blockchain can use smart contracts to enable blockchain participants to realize information exchange and sharing in a conflict-free manner without a trusted third party.

### 2.2. Vector commitment

A cryptographic commitment is a cryptographic primitive. Cryptographic commitment can be thought of as a digitized envelope. The committed message M is equivalent to a letter in an envelope. When you need to expose this information M, you just open the envelope. The commitment scheme is both binding and hiding. Binding means that even the commitmenter cannot change the committed information after the commitment is made. Hiding refers to the fact that no one except the commitmenter can know the committed message before the commitment is opened (Damgård & Fujisaki, 2002; Fan & Zhu, 2019). Vector commitment is a special kind of commitment scheme. It can commit a vector v of length n, and then open the commitment at any position $i \in [n]$. In simple terms, vector commitment is the ability to commit a set of messages at the same time and select a specific message to open. Vector commitment is mainly concerned with binding, and the hiding of vector commitment can be obtained by combining various encryption algorithms (Gorbunov, Reyzin, Wee, & Zhang, 2020; Lai & Malavolta, 2019).

The Merkle tree can be regarded as a vector commitment (Kate et al., 2010). The size of its opening proof is related to the depth of the Merkle tree. As shown in Fig. 3, the leaves of the Merkle tree are the committed vectors and the root of the Merkle tree is the corresponding commitment. All the sibling nodes in the path from the leaf node to the root node and the leaf node itself constitute an opening proof of the vector commitment. (In Fig. 3, $\text{Hash}_{1-8}$ is the commitment value of vector data. $\text{Hash}_3$, $\text{Hash}_4$, $\text{Hash}_{12}$, and $\text{Hash}_{5678}$ are the opening proof of $\text{Data}_3$). In the blockchain system based on the Merkle tree, the opening proof could be used as a storage proof. When verifying the integrity and correctness of a data piece, first use the storage proof to recalculate the Merkle root, and then compare it with the Merkle root in the block header. When both are the same, it could prove that the data piece is stored in the blockchain

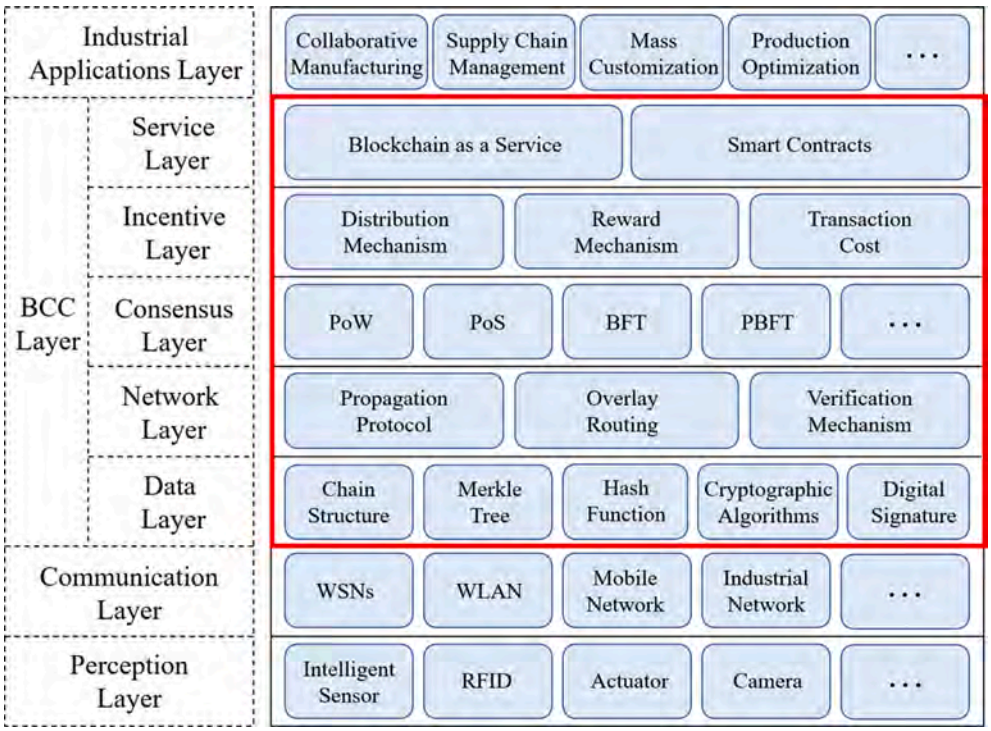



**Fig. 2.** Blockchain-enhanced IIoT architecture.

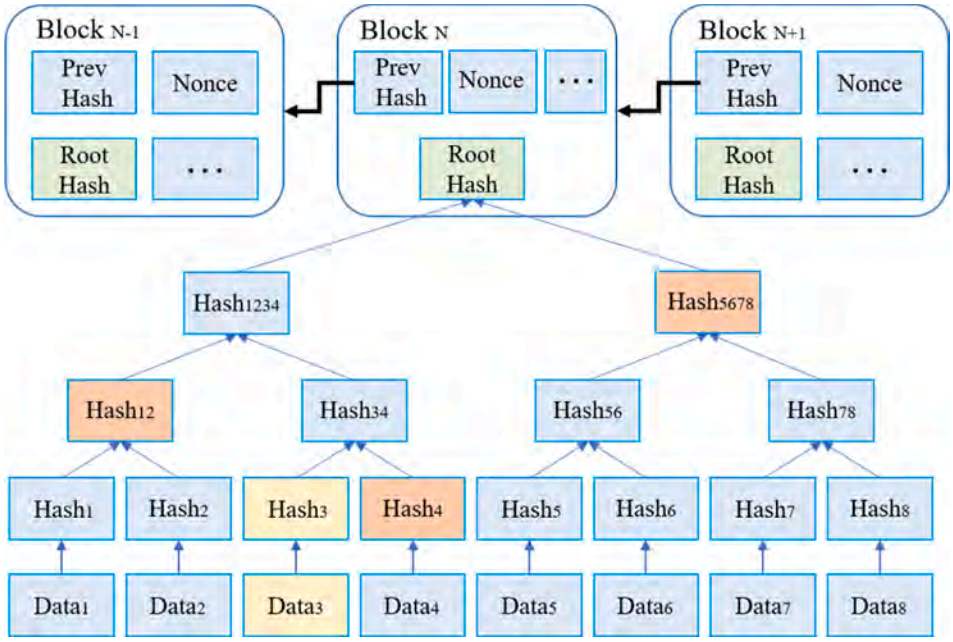



**Fig. 3.** Merkle tree.

(Patel et al., 2017).

Campanelli et al. (2020) proposed the concept of IASVC and constructed two kinds of IASVC based on RSA in the hidden order group. Their schemes can realize unlimited disaggregation and aggregation of proofs and apply it to verifiable distributed storage. IASVC is composed of the following six polynomial time algorithms:

**IASVC.Setup** ($1^{\lambda}$, $M$): Given the security parameter $\lambda$ and the description of the message space $M$, the algorithm will output the public parameters $pp$.

**IASVC.Com** ($pp$, $v$): Given $pp$ and vector $v \in M^n$, the commitment algorithm outputs a commitment $C$ and auxiliary information $aux$.

**IASVC.Open** ($pp$, $I$, $v_I$, $aux$): Given $pp$, vector $v_I$, ordered index set $I \subset N$ and auxiliary information $aux$, the algorithm will output a proof $\pi_I$, where $v_I$ is the $I$ subvector of the commitment message.

**IASVC.Verify** ($pp$, $C$, $I$, $v_I$, $\pi_I$): Given $pp$, the commitment $C$, the ordered set of indices $I \subset N$, the vector $v_I$ and the proof $\pi_I$. The verification algorithm only accepts when $\pi_I$ is a valid proof of $C$ (outputs 1).

**IASVC.Disagg** ($pp$, $I$, $v_I$, $\pi_I$, $K$): Given $pp$, the ordered set of index $I \subset N$, the ordered set of index $K \subset I$, the vector $v_I$ and the proof $\pi_I$, the disaggregation algorithm generates the proof $\pi_K$ of the vector corresponding

to the index set $K$.

**IASVC.Agg** ($pp$, ($I$, $v_I$, $\pi_I$), ($J$, $v_J$, $\pi_J$)): Given $pp$, two triples ($I$, $v_I$, $\pi_I$), ($J$, $v_J$, $\pi_J$), and the aggregation algorithm generates the proof $\pi_K$ of the vector corresponding to the index set $K = I \cup J$.

**Definition 1**. (*Correctness*) For any security parameter λ, any vector length n, any ordered index set $I \in [n]$, satisfying:

$$pr\left[\begin{array}{l} IASVC.Ver(pp, C, I, y, \pi_I) = 1; \\ IASVC.Ver(pp, C, I, y, \pi_D) = 1; \\ IASVC.Ver(pp, C, I, y, \pi_S) = 1; \end{array} \middle| \begin{array}{c} IASVC.Setup(1^{\lambda}, \mathrm{M}) \rightarrow pp; \\ IASVC.Com(\mathrm{pp}, \mathrm{v}) \rightarrow (\mathrm{C}, \mathrm{aux}); \\ IASVC.Open(\mathrm{pp}, \mathrm{I}, \mathrm{y}, \mathrm{aux}) \rightarrow \pi_I; \\ IASVC.Disagg(pp, I, v_I, \pi_I, D) \\ \rightarrow \pi_D; \\ IASVC.Agg(pp, (I, v_I, \pi_I), (J, v_J, \pi_J)) \\ \rightarrow \pi_S; \end{array} \right] \in \mathrm{negl}(\lambda)$$

Then the scheme is correct.

**Definition 2**. (*Location Binding*) If for any PPT adversary A, satisfying:

$$pr\left[\begin{array}{l} IASVC.Ver(pp, C, I, \boldsymbol{v_I}, \pi) = 1 \\ \wedge x \neq \mathrm{x}', where \mathrm{x}' \in I \cap J \\ \wedge IASVC.Ver(pp, C, J, \boldsymbol{v_J}, \pi') = 1 \end{array} \middle| \begin{array}{l} IASVC.Setup(1^{\lambda}, M) \rightarrow pp \\ A(pp) \rightarrow (C, I, \boldsymbol{v_I}, \pi, \boldsymbol{v_J}, \pi') \end{array} \right] \in \mathrm{negl}(\lambda)$$

Then the scheme has location binding.

**Definition 3**. (*Conciseness*) If there is a fixed polynomial $ploy(\lambda)$ in the security parameters, such that the size of commitment $C$, the output of IASVC.Open, IASVC.Disagg and IASVC.Agg are all constrained by $ploy(\lambda)$, then the IASVC is concise.

## 3. The advantages and challenges of blockchain-enabled IIoT

With the introduction of the concept of Industrial 5.0, the technology of IIoT is also facing more challenges. In terms of system operation, the traditional " cyber-physical " system has become a more complex "human-cyber-physical" system, and the number of sensors in the IIoT system has also greatly increased (Nahavandi, 2019; Wang, Gao, Liu, Wu, & Lim, 2019). This brings tremendous management pressure to the centralized management system. In the supply chain, the IIoT will be used for customer personality customization, raw material procurement, inventory management and so on. The information systems of these subjects in supply chain are independent and not connected with each other, and there is the problem of data forgery. In addition, with the addition of a large number of personalized customized users, the non-repudiation of orders is also very important. In terms of data protection, the IIoT has very high requirements for data security (Rani & Kumar, 2017; Wang, Gao, Liu, Sangaiah, & Kim, 2019). The centralized management mode may cause data loss due to individual equipment failures. When the central node is attacked, it may also lead to data leakage or tampering. In the process of data transmission, the security protection of IIoT nodes is very fragile, and there may be threatened by DDoS attacks and cross-heterogeneous network attacks (Ge et al., 2020; Wang, Zou, Lei, Sherratt, & Wang, 2020; Zhou, Guo, & Deng, 2019). Blockchain technology is a useful tool to solve the above problems (Jameel et al., 2020; Zhao et al., 2018).

Although blockchain can bring higher security to the IIoT, there are still deficiencies in communication efficiency, which cannot meet the timeliness requirements of the IIoT. A large amount of data needs to be transmitted in the IIoT. The communication resources are extremely valuable. A tremendous number of large size proof will cause communication congestion, cause network communication delays, seriously affect the stability, efficiency and security of the IIoT, and may cause serious economic losses, so it is necessary to reduce the proof size (Yan et al., 2012).

The existing blockchain systems use Merkle to store data, and a series of hash operations are required to reconstruct the Merkle root when verifying the correctness and integrity of the data piece. The size of storage proof is large, which is affected by the depth of Merkle tree and the number of data pieces to be verified. Moreover, data validation cannot be processed in batches, and a single proof can only validate a single data. This is not conducive to the application of blockchain in the IIoT. In addition, Industrial 5.0 brings "human" into IIoT systems. IIoT will collect all kinds of privacy information of users, so it is necessary to solve the problem of user privacy protection (Hassan, Rehmani, & Chen, 2020; Zhou, Long, Chen, & Yang, 2019). In addition, customers participate in the design process in parallel in age of industrial 5.0. The sensor nodes of the IIoT system will join or exit frequently, and the user's data upload qualification in the system needs to be deleted or added in time. The regulatory node needs to store the list of all the nodes qualified for data upload in the system, which also brings a lot of storage and management pressure to the regulatory node. In order to solve the above problems, we propose a new data storage mechanism based on blockchain.

## 4. The proposed blockchain-based IIoT data storage mechanism

In this section, we propose a blockchain-based data storage mechanism in IIoT. In Section 4.1 we introduced the IASVC-based data storage mechanism. Section 4.2 describes in detail how to replace the data upload qualification list with IASVC. Section 4.3 describes the specific algorithm in detail and proves its correctness and security.

### 4.1. Data storage mechanism based on IASVC in blockchain

Firstly, this paper constructs a data storage Mechanism based on IASVC as shown in Fig. 4. There are three kinds of nodes in this mechanism, namely sensor nodes, storage nodes and regulatory nodes. In our mechanism, the sensing nodes are not various sensors but data aggregators. This is because the underlying sensors do not have powerful computing capabilities and have power limitations. Therefore, the data aggregator needs to first collect the underlying sensor data and then preprocess the data. The regulatory node is used to control the data upload qualification of the sensor nodes. The storage node is responsible for storing data.

In the blockchain data storage mechanism based on the Merkle tree, when verifying the integrity and correctness of the data piece, the hash sequence from the data piece to be verified to the Merkle root should be provided as the storage proof. The storage proof size is related to the number of data pieces to be verified and the Merkle tree depth. In order to reduce the proof size and improve the communication efficiency, this paper proposes to replace the Merkle tree with IASVC, which can provide a constant size storage proof for multiple data pieces. In addition, storage proof can be flexibly disaggregated and aggregated. In our storage mechanism, encryption algorithms can be selected to protect data privacy. Data that does not have a privacy requirement can be not encrypted to increase system efficiency.

This is shown in Fig. 5, the block structure based on IASVC consists of two parts: the block header and the block body. The storage node is responsible for storing all the information of the block header and the block body. The difference is that IASVC replaces the Merkle tree structure of the traditional blockchain in this paper.

Some specific operations in our mechanism are described as follows:

**Setup:** As with other blockchain applications, the sensor node holds a pair of public and private key pairs ($PK$, $SK$). When the sensor node uploads data, it needs to use the private key to sign the transaction, and other nodes use the public key to verify the transaction and determine whether the transaction is issued by the private key holder.

**Data storage transaction:** When the sensor node needs to upload data to the blockchain, the sensor node issues a data upload transaction $Tx = [Data, PK]$, and the data can be encrypted in combination with

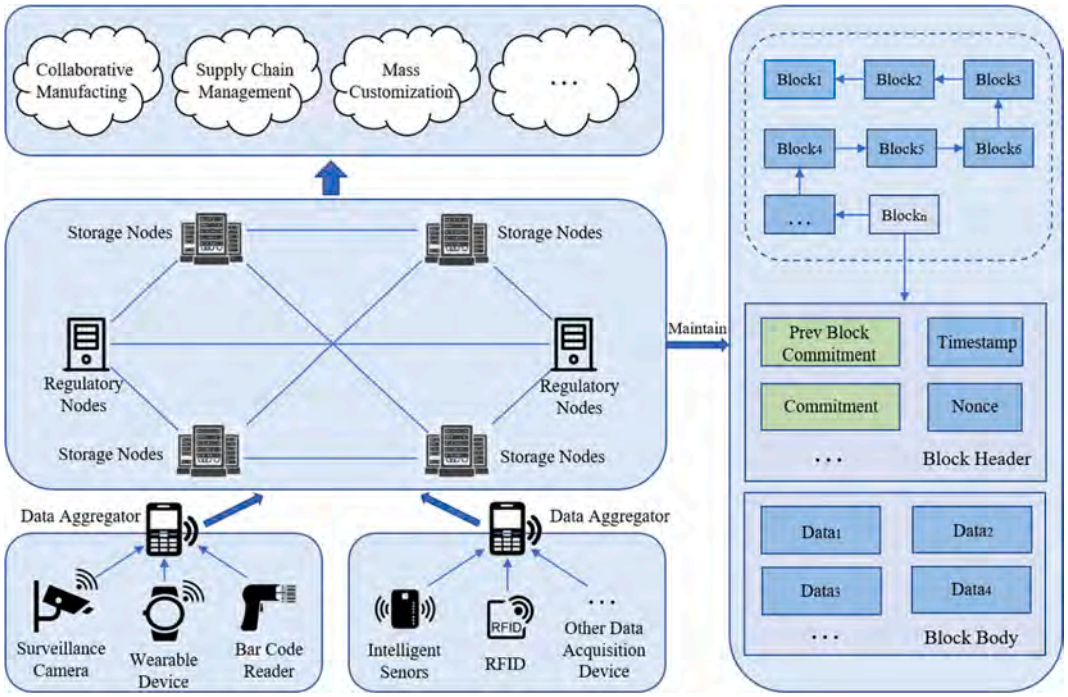



**Fig. 4.** Blockchain data storage mechanism based on IASVC.

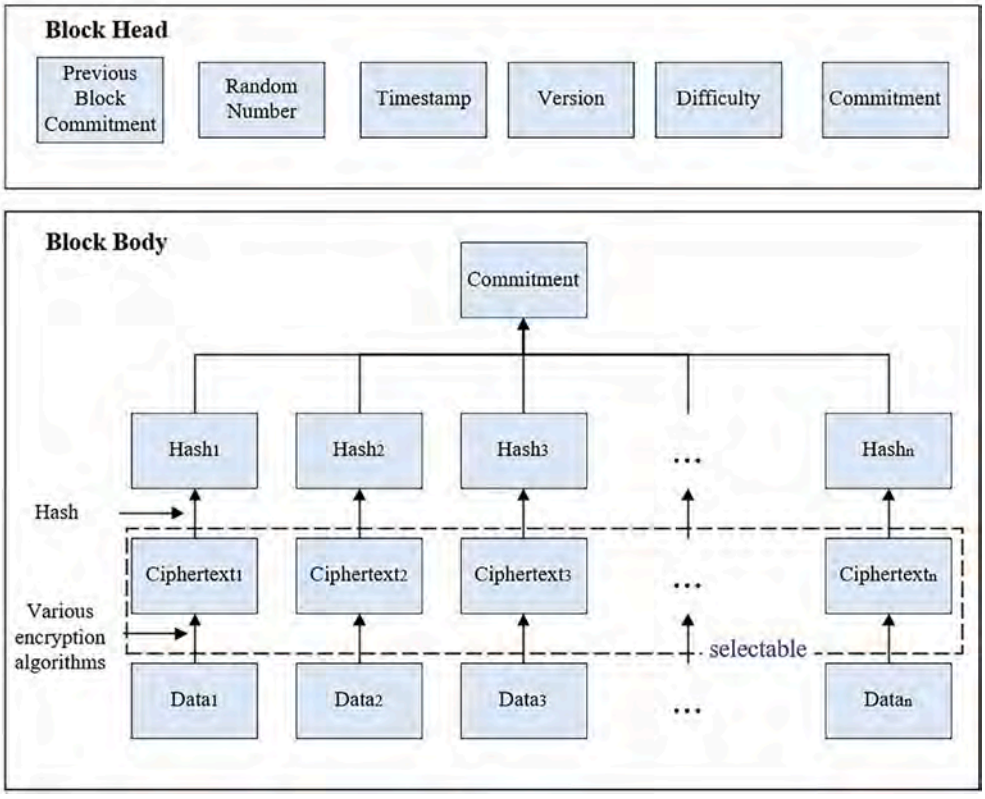



**Fig. 5.** Block structure based on IASVC.

various encryption algorithms to maintain its hidden attributes. The sensor node uses the private key *SK* corresponding to the public key *PK* to sign the transaction, and then broadcast it to the entire blockchain network.

**Block generation:** Blocks are generated by storage nodes, and the storage nodes collect the transactions broadcast in the blockchain network. After the storage nodes collect enough legal data pieces, use the specified hash function to calculate the corresponding hash value for each data piece. All the hash values are organized into vector $[hash_1, \cdots, hash_n]$. Set the index $[1, \cdots, n]$ for the vector, and call IASVC.Com algorithm to commit to the vector. Finally, put the calculated commitment value into the block header, and put all the collected data pieces to be

stored into the block body in. As shown in Fig. 5.

**Data storage proof generation:** When it is necessary to verify the integrity and correctness of the data pieces in the block, the storage node can generate a data storage proof to make users believe that the data pieces to be verified are indeed the data pieces stored in the blockchain. In order to verify the data storage proof, the user needs to store the information of the corresponding block header, the storage node needs to return the corresponding data pieces and the index $I \in [n]$ corresponding to the data pieces, and provide the storage proof of the data pieces. The storage proof is generated by IASVC.Open algorithm. Users can use the commitment value in the block header, the specified hash function and the information returned by the storage node as the input of the IASVC. Ver algorithm to verify the integrity and correctness of the data pieces.

**Disaggregation and aggregation of proof:** The disaggregation function can not only reduce communication consumption, but also meet the user's data privacy needs. After the user obtains the storage proof of the data pieces corresponding to index $I$, the storage proof of the data piece corresponding to the $I$ subset can be provided to other users to reduce the operating pressure of the storage node. When a user with storage proof of data pieces corresponding to index $I$ needs to provide storage proof of data pieces corresponding to index $K$ ($K \subseteqq I$) to other users, the proof of data pieces corresponding to index $K$ can be disaggregated from the proof of data pieces corresponding to index $I$ by IASVC.Disagg algorithm. In addition, multiple proofs can also be aggregated through IASVC.Agg algorithm to reduce the amount of communication.

### 4.2. Data upload qualification verification mechanism base on IASVC

Due to the emergence of mass customization requirements, sensor nodes in the blockchain may frequently join or exit. The regulatory node can flexibly control the joining or exiting of users. In addition, when some sensor nodes are found to be abnormal, the data upload qualification of the problem node can be deleted in time. This paper proposes to use IASVC to set node data upload qualification. When determining the node upload qualification, the storage node does not need to store and search the node qualification list, and only needs to verify a commitment. This is similar to solving the storage pressure on the user balance list by using vector commitment in a stateless blockchain. First of all, we need to construct a commitment of a bit vector for the sensor nodes in the blockchain network (this commitment is named qualification commitment). Its index value is $[1, n]$. The index is mapped to the public key of each data collection node. The value of each vector position is only 0 and 1. A user with a vector position value of 0 is not qualified for data upload, while the user with a vector position value of 1 is qualified for data upload. The sensor node needs to submit the qualification proof when uploading data. The qualification proof is generated by the regulatory node to call the IASVC.Open algorithm (In this application, the input I of the IASVC.Open algorithm is a single element rather than a subset). The submitted qualification proof is verified by the storage node to call the IASVC.Ver algorithm. Only when the corresponding position value of the vector is 1 and the output of IASVC.Ver is 1, the user is eligible for data upload, otherwise the storage nodes will not storage the data piece into Block. Some specific operations are described as follows:

**Setup:** Set a zero vector of length n, and call IASVC.Com algorithm to commit to the zero vector.

**Qualification modification:** There are two types of qualification modification, namely, disqualification and restoration. When a new sensor node enters the blockchain system, a certain index of the vector is assigned to the node, and the value of the corresponding position of the vector is set to 1. Then the qualification proof corresponding to the index is provided to the newly registered node. When it is found that a certain sensor node is at risk, the value of the vector position corresponding to the public key of the sensor node is set to 0. When the vector value changes, the corresponding commitments and qualification proof must be updated accordingly.

**Proof aggregation:** During the communication process, the node can call the IASVC.Agg algorithm to aggregate multiple proofs into one proof to reduce the amount of communication data.

**Commitment update:** The commitment update operation can only be performed by the regulatory node. When the commitment needs to be updated (for example: some sensor nodes maliciously upload data, or some sensor nodes are to be deleted (or added) from the blockchain network, etc.), regulatory node calls the IASVC. UpdataCom algorithm to update commitment. Then, the regulatory node signs the updated commitment with the private key and broadcasts it to the blockchain network. After the storage node receives the signed commitment, it needs to verify the signature with the public key of the regulatory node. If the signature verification is passed, the storage node adds the updated commitment to the block header of the new block.

**Qualification proof update:** After the commitment is updated, the sensor node needs to call the IASVC. UpdataProof algorithm to update its qualification proof.

### 4.3. IASVC based on bilinear mapping

In order to meet the requirements of the above applications, this paper starts from the subvector commitment scheme of Lai and Malavolta (2019), adds proof disaggregation and proof aggregation algorithms to it, expanding it to IASVC. The specific scheme is described as follows:

#### 4.3.1. Preliminaries

**Notation:** This paper, uses λ to denote our security parameters, let $negl(\cdot)$ denote any negligible function, and uses $poly(\cdot)$ to denote any polynomial whose upper bound is a certain univariate polynomial. If algorithm A is modeled as a probabilistic Turing machine running at time $ploy(\lambda)$, it is called a polynomial time algorithm. For a positive integer $n$, we let $[n] := \{1, \cdots, n\}$ denote an index set of size $n$. $v$ represents the vector, $v_I$ represents the subvector corresponding to the $I$ set, and $v_i$ represents the value of the vector at position $i$.

**Bilinear Pairing:** Let $G_1$, $G_2$ be an additive cyclic group of order $p$, $G_T$ is a multiplicative cyclic group of the same order. Define $e: G_1 \times G_2 \rightarrow G_T$ as a bilinear mapping. Where $p$ is a prime number. Let integer $a$, $b \in Z_p^*$, then the bilinear mapping satisfies the following three properties:

**Bilinearity:** For any $P \in G_1$, $Q \in G_2$ and $a, b \in Z_p^*$, $\exists e(P^a, Q^b) = e(P, Q)^{ab}$.

**Non-degeneracy:** $\exists P \in G_1$, Q $\in$ $G_2$, such that $e(P, Q) \neq 1$;

If $g_1$ is the generator of $G_1$ and $g_2$ is the generator of $G_2$, then $e(g_1, g_2)$ is the generator of $G_T$.

**Computability:** For any $P \in G_1$, $Q \in G_2$, there are effective algorithms so that the result of $e(P, Q)$ can be obtained in polynomial time.

When $G_1 = G_2$, the bilinear mapping is said to be symmetric.

**CubeDH (Cube Diffie-Hellman) Assumption:** Let ($P$, $G$, $G_T$, $e$, $g$) be the original set of uniformly randomly generated bilinear pair parameters, where $p$ is a $k$-bit prime. For some $x$ randomly selected from $Z_P$, given the element $g$ and $g^x$ in $G$. It is negligible for adversaries with limited computing power to calculate the probability of $Z$ satisfying $Z = e(g, g)^{x^3}$ in polynomial time.

#### 4.3.2. Specific scheme

**IASVC.Setup**: Generate a bilinear mapping $e: G \times G = G_T$. Let $G_1$, $G_2$ be an additive cyclic group of order $p$, $G_T$ is a multiplicative cyclic group of the same order ; Let $g$ be a random generator, $g \in G$; randomly select $z_1, \cdots, z_q$ from $Z_p$; Calculate $h_i = g^{z_i}$ for all i $\in$ [q], and calculate $h_{i,j} = g^{z_i z_j}$ for all $i,j \in [q]$, $i \neq j$. Set the public parameters $pp = (g, \{h_i\}_{i \in [q]}, \{h_{i,j}\}_{i,j \in [q], i \neq j})$.

**IASVC.Com** ($pp$, $v_{i,i \in [1,q]}$): Given $pp$ and vector $v_{i,i \in [1,q]}$. The algorithm

calculates commitment $C = \prod_{i=1}^{q} h_i^{v_i}$ and auxiliary information $aux = (v_1, \cdots, v_q)$.

**IASVC.Open** ($pp$, $v_I$, $I$, $aux$): Given $pp$, vector $v_I$, ordered index set $I$ and auxiliary information $aux$. The algorithm calculates proof $\pi_I = \prod_{i\in I}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j}$.

**IASVC.Verify** ($pp$, $C$, $\pi_I$, $I$, $v_I$): Given $pp$, the commitment $C$, the ordered index set $I$, the vector $v_I$ and the proof $\pi_I$. The algorithm verifies that $e\left(\frac{C}{\prod_{i\in I} h_i^{v_i}}, \prod_{i\in I} h_i\right) = e(\pi_I, g)$ is true.

**IASVC.Disagg** ($pp$, $I$, $v_I$, $\pi_I$, $K$): Given $pp$, the ordered index set $I$, the ordered index set $K$, the vector $v_I$ and the proof $\pi_I$. Let $L = I\backslash K$, the algorithm calculates proof $\pi_K = \pi_I \frac{\prod_{i\in I\backslash L}\prod_{j\in L} h_{i,j}^{v_j}}{\prod_{i\in L}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j}}$.

**IASVC.Agg** ($pp$, ($I$, $v_I$, $\pi_I$), ($J$, $v_J$, $\pi_J$)): Given $pp$, two triples ($I$, $v_I$, $\pi_I$), ($J$, $v_J$, $\pi_J$). Let $L = I \cap J$, if $L \neq \varnothing$, then let $S = J\backslash L$, call VC.Disagg ($pp$, $J$, $v_J$, $\pi_J$, $S$) to calculate $\pi_S$; otherwise $\pi_S = \pi_J$; The calculation proof is $\pi_K = \pi_{I\cup J} = \pi_I \frac{\prod_{i\in S}\prod_{j\in[q]\backslash(I\cup S)} h_{i,j}^{v_j}}{\prod_{i\in I}\prod_{j\in S} h_{i,j}^{v_j}}$.

The following two algorithms are additionally required in 5.2:

**IASVC.UpdateCom** ($pp$, $C$, $i$, $v_i^{'}$, $v_i$): Given $pp$, the commitment $C$, the index $i$, the message $v_i^{'}$ and $v_i$. The algorithm calculates the updated commitment $C^{'} = C \cdot h_i^{v_i^{'} - v_i}$.

**IASVC.UpdateProof** ($pp$, $\pi_i$, $i^{'}$, $v_{i^{'}}^{'}$, $v_{i^{'}}$): Given $pp$, the proof $\pi_i$, the index $i^{'}$, the message $v_i^{'}$ and $v_i$. The algorithm calculates the updated proof $\pi_i^{'}$. When $i = i^{'}$, $\pi_i^{'} = \pi_i$. When $i \neq i^{'}$: $\pi_i^{'} = \pi_i \cdot h_{i,i^{'}}^{v_{i^{'}}^{'} - v_{i^{'}}}$.

#### 4.3.3. Correctness analysis

In this section, we analyze the correctness of the IASVC.Disagg and IASVC.Agg algorithms.

The correctness of IASVC.Disagg algorithm is verified as follows:

$$\pi_K = \pi_{I\backslash L} = \pi_I \frac{\prod_{i\in I\backslash L}\prod_{j\in L} h_{i,j}^{v_j}}{\prod_{i\in L}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j}}$$

$$= \prod_{i\in I}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j} \frac{\prod_{i\in I\backslash L}\prod_{j\in L} h_{i,j}^{v_j}}{\prod_{i\in L}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j}}$$

$$= \prod_{i\in I\backslash L}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j} \cdot \prod_{i\in I\backslash L}\prod_{j\in L} h_{i,j}^{v_j}$$

$$= \prod_{i\in I\backslash L}\prod_{j\in([q]\backslash I)\cup L} h_{i,j}^{v_j}$$

$$= \prod_{i\in I\backslash L}\prod_{j\in([q]\backslash(L\cup K))\cup L} h_{i,j}^{v_j}$$

$$= \prod_{i\in K}\prod_{j\in[q]\backslash K} h_{i,j}^{v_j}$$

$$= \pi_K$$

The correctness of the IASVC.Agg algorithm is verified as follows:

$$\pi_K = \pi_{I\cup S} = \pi_I \frac{\prod_{i\in S}\prod_{j\in[q]\backslash(I\cup S)} h_{i,j}^{v_j}}{\prod_{i\in I}\prod_{j\in S} h_{i,j}^{v_j}}$$

$$= \prod_{i\in I}\prod_{j\in[q]\backslash I} h_{i,j}^{v_j} \frac{\prod_{i\in S}\prod_{j\in[q]\backslash(I\cup S)} h_{i,j}^{v_j}}{\prod_{i\in I}\prod_{j\in S} h_{i,j}^{v_{i,j}}}$$

$$= \prod_{i\in I}\prod_{j\in[q]\backslash(I\cup S)} h_{i,j}^{v_j} \cdot \prod_{i\in S}\prod_{j\in[q]\backslash(I\cup S)} h_{i,j}^{v_j}$$

$$= \prod_{i\in I\cup S}\prod_{j\in[q]\backslash(I\cup S)} h_{i,j}^{v_j}$$

$$= \prod_{i\in K}\prod_{j\in[q]\backslash K} h_{i,j}^{v_j}$$

$$= \pi_K$$

#### 4.3.4. Security analysis

The security of this scheme is mainly location binding. If the CubeDH assumption is true, then the system has location binding. Then we construct the solution of the CubeDH assumption. Adversary A receives an input ($P$, $G$, $G_T$, $g$, $h$, $e$), for some random number $z$ randomly selected from $Z_P$, we have $h = g^z$, and must output $g^{z^2}$. Randomly select an index $i^*, i^* \in [q]$, and set $h_{i^*} := h$. Symbolically, let $z_{i^*} := z$, $z_{i^*}$ is not known by A. For other indexes $i,j \in [q]\backslash\{i^*\}$, random number $z_i$ randomly selected from $Z_P$ and set $h_i = g^{z_i}$, $h_{i,j} = g^{z_i z_j}$. Set $h_{i,i^*} = h_{i^*,i} = g^{z\cdot z_j}$ for each $i \in [q]\{i^*\}$. Set public parameters $pp = (P, G, G_T, g, \{h_i\}_{i\in[q]}, \{h_{i,j}\}_{i,j\in[q],i\neq j}, e)$. A runs B on the input ($1^{\lambda}$, $pp$), then adversary A will output with negligible probability ($C, I, y, \pi, J, y^{'}, \pi^{'}$) such that IASVC.Ver($pp, C, I, y, \pi$) $= 1 \wedge y \neq y^{'} \wedge$ IASVC.Ver($pp, C, J, y^{'}, \pi^{'}$) $= 1$. Since $y \neq y^{'}$, there is at least one set ($x \in y$, $x^{'} \in y^{'}$) that satisfies $x \neq x^{'}$. Under the above conditions, we have a probability of $1/q$ to obtain $x_{i^*} \neq x_{i^*}^{'}$. Through the verification algorithm, we have:

$$e\left(\frac{C}{\prod_{i\in I} h_i^{x_i}}, \prod_{i\in I} h_i\right)^{\sum_{i\in J} z_i} \cdot e\left(\pi_I, \prod_{i\in J} h_i\right) = e\left(\frac{C}{\prod_{i\in J} h_i^{x_i^{'}}}, \prod_{i\in J} h_i\right)^{\sum_{i\in I} z_i} \cdot e\left(\pi_J, \prod_{i\in I} h_i\right) \quad (1)$$

Since $e\left(C, \prod_{i\in I} h_i\right)^{\sum_{i\in J} z_i} = e\left(C, \prod_{i\in J} h_i\right)^{\sum_{i\in I} z_i}$.

Equation (1) can be rewritten as:

$$e\left(\prod_{i\in I} h_i^{x_i}, \prod_{i\in I} h_i\right)^{\sum_{i\in J} z_i} \cdot e\left(\pi_I, \prod_{i\in J} h_i\right) = e\left(\prod_{i\in J} h_i^{x_i^{'}}, \prod_{i\in J} h_i\right)^{\sum_{i\in I} z_i} \cdot e\left(\pi_J, \prod_{i\in I} h_i\right) \quad (2)$$

Rewrite equation (2) further as:

$$e(g,g)^{(\sum_{i\in I} z_i x_i)(\sum_{i\in I} z_i)(\sum_{i\in J} z_i)} \cdot e\left(\pi_I, \prod_{i\in J} h_i\right) = e(g,g)^{(\sum_{i\in J} z_i x_i')(\sum_{i\in J} z_i)(\sum_{i\in I} z_i)} \cdot e\left(\pi_J, \prod_{i\in I} h_i\right) \quad (3)$$

The left index part:

$$\left(\sum_{i\in I} z_i x_i\right)\left(\sum_{i\in I} z_i\right)\left(\sum_{i\in J} z_i\right) = x_{i^*} z_{i^*}^3 + x_{i^*} z_{i^*}^2 \left(\sum_{i\in J\setminus\{i^*\}} z_i\right) + x_{i^*} z_{i^*} \left(\sum_{i\in I\setminus\{i^*\}} z_i\right)\left(\sum_{i\in J} z_i\right) + \left(\sum_{i\in I\setminus\{i^*\}} z_i x_i\right)\left(\sum_{i\in I} z_i\right)\left(\sum_{i\in J} z_i\right) = x_{i^*} z_{i^*}^3 + \alpha$$

The right index part:

$$\left(\sum_{i\in J} z_i x_i'\right)\left(\sum_{i\in J} z_i\right)\left(\sum_{i\in I} z_i\right) = x_{i^*}' z_{i^*}^3 + x_{i^*}' z_{i^*}^2 \left(\sum_{i\in I\setminus\{i^*\}} z_i\right) + x_{i^*}' z_{i^*} \left(\sum_{i\in J\setminus\{i^*\}} z_i\right)\left(\sum_{i\in I} z_i\right) + \left(\sum_{i\in J\setminus\{i^*\}} z_i x_i'\right)\left(\sum_{i\in J} z_i\right)\left(\sum_{i\in I} z_i\right) = x_{i^*}' z_{i^*}^3 + \beta$$

Observation shows that in the variable $z_{i^*}$, $\alpha$ and $\beta$ are at most quadratic. We can rewrite equation (3) as:

$$e(g,g)^{x_{i^*} z_{i^*}^3 + \alpha} \cdot e\left(\pi_I, \prod_{i\in J} h_i\right) = e(g,g)^{x_{i^*}' z_{i^*}^3 + \beta} \cdot e\left(\pi_J, \prod_{i\in I} h_i\right) \quad (4)$$

Rewrite equation (4) further as:

$$e(g,g)^{(x_{i^*} - x_{i^*}') z_{i^*}^3} = e\left(\pi_J, \prod_{i\in I} h_i\right) \cdot e\left(\pi_I, \prod_{i\in J} h_i\right)^{-1} \cdot h^{\beta-\alpha} \quad (5)$$

Since $x_{i^*} \neq x_{i^*}'$, the left side of the equation is not 0, and the right side of the equation can be calculated by the method of order reduction. Therefore, the CubeDH assumption is broken. Since the CubeDH assumption is difficult, the probability of the scheme being broken is negligible.

## 5. Performance analysis and comparison

In this section, we analyze the proposed scheme and compare the commitment schemes presented in this paper asymptotically with other commitment schemes in Table 1. The schemes we compared include the Merkle tree, CFGKN 13 (Catalano & Fiore, 2013) and CF 20 (Campanelli et al., 2020). Where N represents the number of transactions stored in a single block, and L represents the number of data pieces that need to be validated. The complexity in this table is asymptotic in terms of the number of exponents, pairings and field operations. Com represents the time to compute the commitment. I-Open represents the time to compute the proof corresponding to subset I, and I-Ver is the verification overhead of the proof corresponding to subset I. Disagg refers to whether the scheme has disaggregation algorithm, and Agg refers to whether the scheme has aggregation algorithm.

Then we analyzed and compared the proposed scheme with other schemes in terms of communication consumption. In our experimental setting, we first set the size of each data piece as 64-bit. Then we set the security level to 128-bit security. We use 2048-bit class group for hidden order group scheme, and BLS12-381 for bilinear group scheme. Fig. 6 shows the proof size relative to the number of data pieces L. The four subgraphs (a), (b), (c) and (d) respectively show the experimental results when 500, 1000, 2000, 5000 transactions are stored in a single block.

**Table 1**
Performance comparison of schemes.

| Scheme | Com | I-Open | I-Ver | Disagg | Agg |
|---|---|---|---|---|---|
| Merkle | O(N) | — | — | — | — |
| CFGKN 13 | O(N) | — | — | — | — |
| CF 20 | O(N·logN) | O(N·logN) | O(L·logL) | √ | √ |
| This work | O(N) | O(L·N) | O(L) | √ | √ |

— means does not have the function.
√ means has the function.

Fig. 6 measures the proof size when verifying the availability of data piece. Since the size of the Merkle proof is affected by the depth of the Merkle tree, the proof size also grows as the number of transactions stored per block grows. However, the proof size of single data piece in vector commitment scheme is not affected by the number of transactions stored in per block. Under the setting of 500TXS, 1000TXS, 2000TXS and 5000TXS, the proof sizes of single data piece in this scheme are 16.7%, 15.0%, 13.6% and 11.5% of the Merkle tree scheme, respectively. As the number of transactions increases, the advantages of our scheme become more significant. Since the Merkle tree scheme and the LM10 scheme have no aggregation algorithm, the size of proof is not fixed. As L increases, the proof sizes of Merkle tree scheme and LM scheme also corresponding increase. Our scheme and the CF20 scheme have aggregation algorithm. As L increases, they have proof of fixed size. But the scheme in this paper is a bilinear group scheme. It has a smaller proof size under the same security level.

Fig. 7 measures the total communication consumption when verifying the availability of data pieces. The cost of verifying data availability mainly includes:$|C| + L\cdot|\pi| + L\cdot|M| + L\cdot|I|$. Where $L$ is the number of data pieces, $|\pi|$ is the size of proof,$|C|$ is the size of commitment, $|M|$ is the size of data pieces, and $|I|$ is the size of index.

In the total communication consumption, $|M|$ is inherent consumption. The communication consumption difference between schemes depends on the $|\pi|$, $|C|$, and $|I|$. As L increases, the advantages of schemes with aggregation function gradually increase. In the case of 1000TXs, the total communication consumption when verifying 1, 2, 3, 4, and 5 data pieces are respectively 79.5%, 66.6%, 64.9%, 64.1, 63.6% of the Merkle scheme. In the setting of 500TXs, 1000TXs, 2000TXs and 5000TXs, our scheme has the minimum communication consumption, and the advantage of the scheme increases with the increase of the number of TXs processed in each block.

## 6. Conclusion and future work

In this paper, we use the IASVC to improve the traditional blockchain

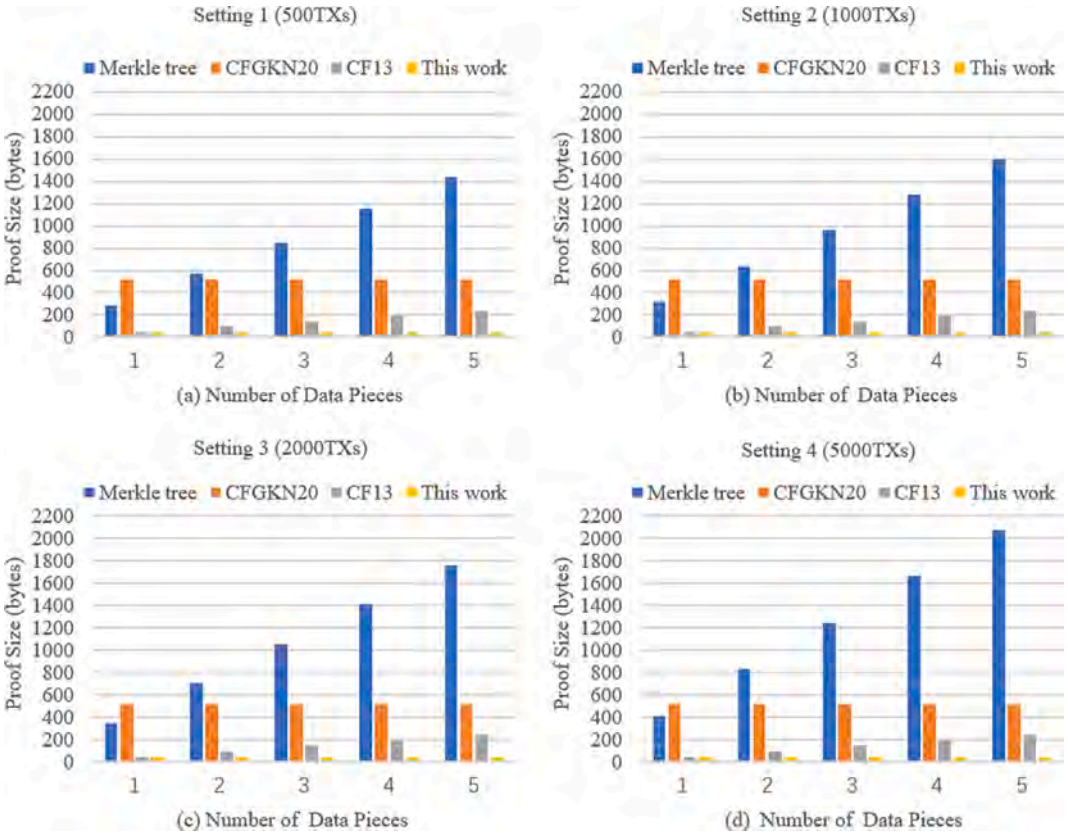



**Fig. 6.** Comparison of Proof Size.

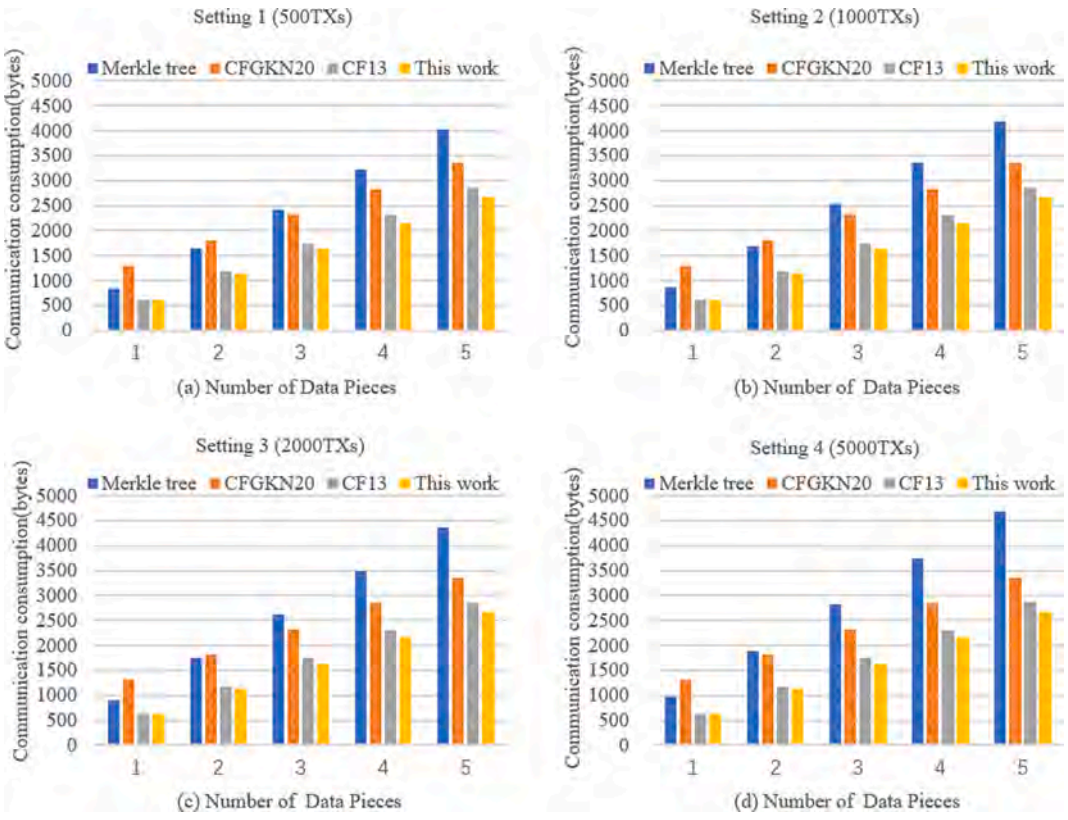



**Fig. 7.** Comparison of Total Communication Consumption.

system, and use the improved blockchain system to protect IIoT data security. The traditional blockchain system uses Merkle trees to store data. The size of proof in verifying the correctness and completeness of the data is large. Our scheme significantly reduces the size of proof, improves the efficiency of communication, and could flexibly combine various encryption algorithms to protect the privacy of IIoT data. We also use the IASVC to set the IIoT node data upload qualification to reduce the node storage pressure. Then we construct an IASVC based on bilinear mapping to meet the above application requirements. Finally, we compared our scheme with other schemes. According to comparison, we found that our scheme not only has a smaller proof size for a single data piece, but also the smallest proof size for multiple data pieces. Our scheme can effectively reduce communication consumption and improve communication efficiency.

In this work, we only considered the proof aggregation of a single commitment. A natural extension is to aggregate proof of multiple commitments, which can further reduce communication consumption and improve communication efficiency. We will build vector commitment with the ability to aggregate proofs of different commitments as our first future work. Multilinear mapping is a new concept in recent years. It has much more power than bilinear mapping. We will use multilinear mapping to construct IASVC as our second future work.

## Funding

This work is supported by the NSFC [61772280, 62072249, 61772454, 62072056]. This work was funded by the Researchers Supporting Project No. (RSP-2021/102) King Saud University, Riyadh, Saudi Arabia.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

Alladi, T., Chamola, V., Parizi, R. M., & Choo, K. K. R. (2019). Blockchain Applications for Industry 4.0 and Industrial IoT: A Review. *IEEE Access, 7*, 176935–176951. https://doi.org/10.1109/ACCESS.2019.2956748

Aslam, F., Aimin, W., Li, M., & Rehman, K. U. (2020). Innovation in the era of IoT and industry 5.0: Absolute innovation management (AIM) framework. *Information (Switzerland), 11*(2). https://doi.org/10.3390/info11020124

Boneh, D., Bünz, B., & Fisch, B. (2019). Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains. In *Annual International Cryptology Conference, Santa Barbara, CA*. https://doi.org/10.1007/978-3-030-26948-7_20

Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Computers in Industry, 101*, 1–12. https://doi.org/10.1016/j.compind.2018.04.015

Campanelli, M., Fiore, D., Greco, N., Kolonelos, D., & Nizzardo, L. (2020). Incrementally Aggregatable Vector Commitments and Applications to Verifiable Decentralized Storage. In *International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, Korea*. https://doi.org/10.1007/978-3-030-64834-3_1

Catalano, D., & Fiore, D. (2013). Vector commitments and their applications. In *IACR International Conference on Public-Key Cryptography, Nara, Japan*. https://doi.org/10.1007/978-3-642-36362-7_5

Catalano, D., Di Raimondo, M., Fiore, D., & Messina, M. (2011). Zero-knowledge sets with short proofs. *IEEE Transactions on Information Theory, 57*(4), 2488–2502. https://doi.org/10.1109/TIT.2011.2112150

Damgård, I., & Fujisaki, E. (2002). A statistically-hiding integer commitment scheme based on groups with hidden order. *International Conference on the Theory and Application of Cryptology and Information Security*. Berlin: Springer.

Demir, K. A., Döven, G., & Sezen, B. (2019). Industry 5.0 and Human-Robot Co-working. *Procedia Computer Science, 158*, 688–695. https://doi.org/10.1016/j.procs.2019.09.104

Fan, H., & Zhu, H. (2019). Motion vector detection based on local autocorrelation coefficient. *Cluster Computing, 22*, 11633–11639. https://doi.org/10.1007/s10586-017-1428-9

Faruqi, U. A. (2019). Future Service in Industry 5.0: Survey Paper. *Journal of Symbolic Computation, 2*(1), 67–79. https://doi.org/10.37396/jsc.v2i1.21

Fernández-Caramés, T. M., Blanco-Novoa, O., Suárez-Albela, M., & Fraga-Lamas, P. (2018). A UAV and blockchain-based system for industry 4.0 inventory and traceability applications. *Multidisciplinary Digital Publishing Institute Proceedings, 4*(1), 26. https://doi.org/10.3390/ecsa-5-05758

Ge, C., Liu, Z., Xia, J., & Liming, F. (2019). Revocable Identity-Based Broadcast Proxy Re-encryption for Data Sharing in Clouds. *IEEE Transactions on Dependable and Secure Computing, 5971*. https://doi.org/10.1109/tdsc.2019.2899300

Ge, C., Susilo, W., Liu, Z., Xia, J., Szalachowski, P., & Liming, F. (2020). Secure Keyword Search and Data Sharing Mechanism for Cloud Computing. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/tdsc.2020.2963978

Gong, L., Yang, B., Xue, T., Chen, J., & Wang, W. (2018). Secure rational numbers equivalence test based on threshold cryptosystem with rational numbers. *Information Sciences, 466*(2016), 44–54. https://doi.org/10.1016/j.ins.2018.07.046

Gorbunov, S., Reyzin, L., Wee, H., & Zhang, Z. (2020). Pointproofs: Aggregating Proofs for Multiple Vector Commitments. In *Proceedings of the ACM Conference on Computer and Communications Security, Virtual Event, USA*. https://doi.org/10.1145/3372297.3417244

Hassan, M. U., Rehmani, M. H., & Chen, J. (2020). Differential Privacy Techniques for Cyber Physical Systems: A Survey. *IEEE Communications Surveys and Tutorials, 22*(1), 746–789. https://doi.org/10.1109/COMST.2019.2944748

Huang, J., Kong, L., Chen, G., Wu, M. Y., Liu, X., & Zeng, P. (2019). Towards secure industrial iot: Blockchain system with credit-based consensus mechanism. *IEEE Transactions on Industrial Informatics, 15*(6), 3680–3689. https://doi.org/10.1109/TII.2019.2903342

Jameel, F., Javaid, U., Khan, W. U., Aman, M. N., Pervaiz, H., & Jäntti, R. (2020). Reinforcement learning in blockchain-enabled IIoT networks: A survey of recent advances and open challenges. *Sustainability (Switzerland), 12*(12), 1–22. https://doi.org/10.3390/su12125161

Kate, A., Zaverucha, G. M., & Goldberg, I. (2010). Constant-size commitments to polynomials and their applications. In *International Conference on the Theory and Application of Cryptology and Information Security, Singapore, Singapore*. https://doi.org/10.1007/978-3-642-17373-8_11

Khattak, H. A., Shah, M. A., Khan, S., Ali, I., & Imran, M. (2019). Perception layer security in Internet of Things. *Future Generation Computer Systems, 100*, 144–164.

Lai, R. W. F., & Malavolta, G. (2019). Subvector Commitments with Application to Succinct Arguments. In *Annual International Cryptology Conference, Santa Barbara, CA*. https://doi.org/10.1007/978-3-030-26948-7_19

Libert, B., & Yung, M. (2010). Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In *Theory of Cryptography Conference, Zurich, Switzerland*. https://doi.org/10.1007/978-3-642-11799-2_30

Lin, Q., Wang, H., Pei, X., & Wang, J. (2019). Food safety traceability system based on blockchain and EPCIS. *IEEE Access, 7*, 20698–20707.

Mao, Y., Zhang, J., Qi, H., & Wang, L. (2019). DNN-MVL: DNN-multi-view-learning-based recover block missing data in a dam safety monitoring system. *Sensors (Switzerland), 19*(13). https://doi.org/10.3390/s19132895

Mondal, S., Wijewardena, K. P., Karuppuswami, S., Kriti, N., Kumar, D., & Chahal, P. (2019). Blockchain inspired RFID-based information architecture for food supply chain. *IEEE Internet of Things Journal, 6*(3), 5803–5813. https://doi.org/10.1109/JIOT.2019.2907658

Nahavandi, S. (2019). Industry 5.0-a human-centric solution. *Sustainability (Switzerland), 11*(16). https://doi.org/10.3390/su11164371

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business and Information Systems Engineering, 59*(3), 183–187. https://doi.org/10.1007/s12599-017-0467-3

Özdemir, V. N. H. (2018). Birth of industry 5.0: Making sense of big data with artificial intelligence, "the internet of things" and next-generation technology policy. *Omics: A Journal of Integrative Biology, 22*(1), 65–76.

Patel, D., Bothra, J., & Patel, V. (2017). Blockchain exhumed. *2017 ISEA Asia Security and Privacy (ISEASP), Surat, India*.

Pieroni, A., Scarpato, N., Di Nunzio, L., Fallucchi, F., & Raso, M. (2018). Smarter City: Smart energy grid based on Blockchain technology. *International Journal on Advanced Science, Engineering and Information Technology, 8*(1), 298–306. https://doi.org/10.18517/ijaseit.8.1.4954

Raikwar, M., Gligoroski, D., & Kralevska, K. (2019). SoK of used cryptography in blockchain. *IEEE Access, 7*, 148550–148575.

Rani, A., & Kumar, S. (2017). A survey of security in wireless sensor networks. In *In 2017 3rd International Conference on Computational Intelligence & Communication Technology*. https://doi.org/10.1109/CIACT.2017.7977334

Ren, Y., Leng, Y., Cheng, Y., & Wang, J. (2019). Secure data storage based on blockchain and coding in edge computing. *Mathematical Biosciences and Engineering, 16*(4), 1874–1892. https://doi.org/10.3934/mbe.2019091

Ren, Y., Leng, Y., Qi, J., Sharma, P. K., Wang, J., Almakhadmeh, Z., & Tolba, A. (2021). Multiple cloud storage mechanism based on blockchain in smart homes. *Future Generation Computer Systems, 115*, 304–313. https://doi.org/10.1016/j.future.2020.09.019

Ren, Y., Leng, Y., Zhu, F., Wang, J., & Kim, H. J. (2019). Data storage mechanism based on blockchain with privacy protection in wireless body area network. *Sensors (Switzerland), 19*(10), 1–16. https://doi.org/10.3390/s19102395

Ren, Y., Liu, Y., Ji, S., Sangaiah, A. K., & Wang, J. (2018). Incentive Mechanism of Data Storage Based on Blockchain for Wireless Sensor Networks. *Mobile Information Systems, 2018*. https://doi.org/10.1155/2018/6874158

Ren, Y., Qi, J., Cheng, Y., Wang, J., & Alfarraj, O. (2020). Digital continuity guarantee approach of electronic record based on data quality theory. *Computers, Materials and Continua, 63*(3), 1471–1483. https://doi.org/10.32604/CMC.2020.06745

Ren, Y., Zhu, F., Sharma, P. K., Wang, T., Wang, J., Alfarraj, O., & Tolba, A. (2020). Data query mechanism based on hash computing power of blockchain in internet of things. *Sensors (Switzerland), 20*(1). https://doi.org/10.3390/s20010207

Salahuddin, M. A., Al-Fuqaha, A., Guizani, M., Shuaib, K., & Sallabi, F. (2018). Softwarization of internet of things infrastructure for secure and smart healthcare. *Computer, 50*(7), 74–79. https://doi.org/10.1109/MC.2017.195

Sharma, P. K., Kumar, N., & Park, J. H. (2018). Blockchain-based distributed framework for automotive industry in a smart city. *IEEE Transactions on Industrial Informatics, 15*(7), 4197–4205. https://doi.org/10.1109/TII.2018.2887101

Skobelev, & Borovik. (2017). On the way from Industry 4.0 to Industry 5.0: From Digital Manufacturing to Digital Society. *International Scientific Journal "Industry 4.0", 2*(6), 307–311.

Tomescu, A., Abraham, I., Buterin, V., Drake, J., Feist, D., & Khovratovich, D. (2020). Aggregatable subvector commitments for stateless cryptocurrencies. In *International Conference on Security and Cryptography for Networks, Amalfi, Italy*. https://doi.org/10.1007/978-3-030-57990-6_3

Wang, J., Gao, Y., Liu, W., Sangaiah, A. K., & Kim, H. J. (2019). An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks. *International Journal of Distributed Sensor Networks, 15*(3). https://doi.org/10.1177/1550147719839581

Wang, J., Gao, Y., Liu, W., Wu, W., & Lim, S. J. (2019). An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks. *Computers, Materials and Continua, 58*(3), 711–725. https://doi.org/10.32604/cmc.2019.05450

Wang, J., Gao, Y., Yin, X., Li, F., & Kim, H. J. (2018). An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks. *Wireless Communications and Mobile Computing, 2018*. https://doi.org/10.1155/2018/9472075

Wang, J., Gao, Y., Zhou, C., Simon Sherratt, R., & Wang, L. (2020). Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs. *Computers, Materials and Continua, 62*(2), 695–711. https://doi.org/10.32604/cmc.2020.08674

Wang, J., Gu, X., Liu, W., Sangaiah, A. K., & Kim, H. J. (2019). An empower hamilton loop based data collection algorithm with mobile agent for WSNs. *Human-Centric Computing and Information Sciences, 9*(1). https://doi.org/10.1186/s13673-019-0179-4

Wang, J., Ju, C., Gao, Y., Sangaiah, A. K., & Kim, G. J. (2018). A PSO based energy efficient coverage control algorithm for wireless sensor networks. *Computers, Materials and Continua, 56*(3), 433–446. https://doi.org/10.3970/cmc.2018.04132

Wang, J., Yang, Y., Wang, T., Sherratt, R. S., & Zhang, J. (2020). Big Data Service Architecture: A Survey. *Journal of Internet Technology, 21*(2), 393–405. https://doi.org/10.3966/160792642020032102008

Wang, J., Zou, Y., Lei, P., Sherratt, R. S., & Wang, L. (2020). Research on recurrent neural network based crack opening prediction of concrete dam. *Journal of Internet Technology, 21*(4), 1161–1169. https://doi.org/10.3966/160792642020072104024

Wang, K., Wang, Y., Sun, Y., Guo, S., & Wu, J. (2016). Green industrial Internet of Things architecture: An energy-efficient perspective. *IEEE Communications Magazine, 54*(12), 48–54.

Xu, J., Zhang, Y., Fu, K., & Peng, S. (2019). SGx-based secure indexing system. *IEEE Access, 7*(1), 77923–77931. https://doi.org/10.1109/ACCESS.2019.2921223

Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on cyber security for smart grid communications. *IEEE Communications Surveys and Tutorials, 14*(4), 998–1010. https://doi.org/10.1109/SURV.2012.010912.00035

Zhao, W., Liu, J., Guo, H., & Hara, T. (2018). ETC-IoT: Edge-Node-Assisted Transmitting for the Cloud-Centric Internet of Things. *IEEE Network, 32*(3), 101–107. https://doi.org/10.1109/MNET.2018.1700164

Zhou, L., Guo, H., & Deng, G. (2019). A fog computing based approach to DDoS mitigation in IIoT systems. *Computers and Security, 85*, 51–62. https://doi.org/10.1016/j.cose.2019.04.017

Zhou, Y., Long, X., Chen, L., & Yang, Z. (2019). Conditional privacy-preserving authentication and key agreement scheme for roaming services in VANETs. *Journal of Information Security and Applications, 47*, 295–301. https://doi.org/10.1016/j.jisa.2019.05.018

Zhou, Y., Zhao, X., Liu, S., Long, X., & Luo, W. (2019). A time-aware searchable encryption scheme for EHRs. *Digital Communications and Networks, 5*(3), 170–175. https://doi.org/10.1016/j.dcan.2018.09.003