Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8482415, 16 pages
https://doi.org/10.1155/2022/8482415

Research Article

A Data Management Model for Intelligent Water Project Construction Based on Blockchain

Zhoukai Wang[1,2] **Kening Wang**[3] **Yichuan Wang**[1,2] **and Zheng Wen**[4]

[1]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 710048, China*
[2]*Shaanxi Provincial Key Laboratory of Network Computing and Security Technology, Xi'an, 710048, China*
[3]*School of Automation and Information Engineering, Xi'an University of Technology, Xi'an, 710048, China*
[4]*School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan*

Correspondence should be addressed to Zhoukai Wang; zkwang@xaut.edu.cn

Received 7 December 2021; Accepted 16 February 2022; Published 9 March 2022

Academic Editor: Qingqi Pei

The engineering construction-related data is essential for evaluating and tracing project quality in industry 4.0. Specifically, the preservation of the information is of great significance to the safety of intelligent water projects. This paper proposes a blockchain-based data management model for intelligent water projects to achieve standardization management and long-term preservation of archives. Based on studying the concrete production process in water conservancy project construction, we first build a behavioral model and the corresponding role assignment strategy to describe the standardized production process. Then, a distributed blockchain data structure for storing the production-related files is designed according to the model and strategy. In addition, to provide trust repository and transfer on the construction data, an intelligent keyless signature based on edge computing is employed to manage the data's entry, modification, and approval. Finally, standardized and secure information is uploaded onto the blockchain to supervise intelligent water project construction quality and safety effectively. The experiments showed that the proposed model reduced the time and labor cost when generating the production data and ensured the security and traceability of the electronic archiving of the documents. Blockchain and intelligent keyless signatures jointly provide new data sharing and trading methods in intelligent water systems.

1. Introduction

In the water conservancy project management, archives have the characteristics of large numbers and comprehensive coverage, and they play an essential role in all aspects of engineering construction. With the increasing investment of water conservancy projects, the scale gradually grows, and the project gradually becomes complex. The management of water conservancy project archives also faces more and more problems, which restrict the development of water conservancy projects. On the other side, the traditional file management mode can no longer adapt to the rapidly developing economic needs, so the introduction of digital archives for water conservancy projects has become an inevitable trend [1, 2, 3]. However, because the construction of water conservancy projects requires the global deployment and management of various units and resources, making the digitization process of its archives difficult, the status of library management leading to the water conservation institutions requires acceleration transformation [4].

At present, digital archives of water conservancy projects have less relevant research in foreign countries, and the research in China is also in the initial stage [5, 6]. Although the new "Archives Law of the People's Republic of China" provides legal and policy guarantees for the informationization of construction files of water conservancy project construction, the relevant research and application are still focused on the initial stage of construction [7]. Other important aspects of water conservancy project construction, such as concrete production and mixing, and metal structure installation, still lack effective information management means [8]. In addition, the current digital file management

methods are relatively simple, with the drawbacks of poor antitampering and antirepudiation capabilities, and their application range is also limited. In total, the current digital file management methods cannot undertake the engineering construction works involving significant safety needs [9].

In response to the shortcomings of traditional file management methods, this paper introduces blockchain and keyless signature techniques [10], takes the concrete mixing process as the research object, and conducts research on data management in intelligent water conservancy construction. The main contributions of this paper are demonstrated as follows.

(1) By employing the smart keyless signatures, this paper established a paperless concrete production and operation management model to monitor the concrete mixing process and prevent data tampering during the process

(2) With the help of the consortium blockchain, this paper built up an intelligent document storage method to effectively supervise the progress, quality, and safety of concrete production and then explore general methods for encryption, storage, and traceability of production files

(3) Integrated with the corresponding model and method, this paper proposed a blockchain-based file management system for concrete mixing procedures and then implemented it in the Hanjiang to Weihe River Project to improve the production management capacity markedly

2. Motivation

Concrete mixing is a vital link in the construction of water conservancy projects, and many engineering archival documents are generated during the mixing process to record the concrete mixing details [11]. These files are crucial basic information for project quality control and problem tracing and are related to the whole life cycle safety of the project. However, the management of concrete production files has problems such as low informationization and insufficient security at present [12, 13]. Firstly, the current management method wastes paper. The volume of files related to concrete mixing production is enormous. The amount of grouting required for reservoir construction is usually more than 100,000 cubic meters, which will generate a massive amount of paper data that is difficult to store and manage. Secondly, the paper-based management method has less credibility. The manually dumped paper files are not standardized, and falsification of the paper files often occurs. Thirdly, the traceability of the paper files is feeble. Currently, the cataloging and archiving of concrete production files have not yet formed a strict and complete discipline and management system. Therefore, it is difficult to achieve practical traceability issues tracing. At last, the current management methods obtain insufficient security since the lack of security and confidentiality control measures for the massive paper files.

In response to the above problems, more and more researchers have devoted their efforts to studying the digital file management of water conservancy projects, especially for the informatization of the concrete mixing process, and preliminary research results have been achieved. The representative projects include the Jingtaichuan Dam in Gansu Province, the Daxing Water Conservancy Hub Project in Guizhou Province, and Chushandian Reservoir in Henan Province [14]. However, these research results still fail to completely solve the shortcomings of low antitampering ability and poor antirepudiation ability of digitized archives [15]. Digital archives are still exposed to risks, and they are difficult to effectively manage the concrete mixing procedure and ensure the procedure's safety.

The quality of concrete production and the management of related production documents are closely associated with the safety of people's lives and property. They have a high level of tamper-proof and repudiation-proof requirements [16]. Although the Chinese government has established the corresponding laws to push forward electronic signatures steadily, digital files are bound to be severe trust and security concerns when transmitted over the Internet and stored in centralized servers for long periods [17]. The higher the sensitivity of the data, the greater the risk of using high-tech means to "blacken" it. Apparently, there are substantial technical difficulties in achieving the highly informative management of concrete mixing files paperless. How to help the concrete mixing files and the corresponding data get rid of the security threat becomes a hot topic in the intelligent water conservancy field.

In 2009, blockchain technology was proposed to guarantee the security of data. Recently, applications based on blockchain have increasingly appeared in various fields of daily social life, such as finance, public services, culture and entertainment, data insurance, and general welfare [18–20]. However, the present archival research on the blockchain mainly focuses on the feasibility of document archive management and specific application methods [21]. Many scholars have proposed their application for standard archival management based on blockchains, such as museum archives, student archives, and medical information archives [22, 23]. Other scholars have also discussed the challenges and troubles that blockchain technology may face when applied in archival management [24, 25]. But there are only a few cases of the practical application of blockchain-based archive management in hydraulic engineering fields [26]. In summary, applying blockchain and related technologies in engineering construction archive management, especially to critical aspects such as concrete production, has received less attention from relevant studies domestic and abroad.

Based on the current research foundation in related fields, this paper takes the whole concrete production process as the research object, integrates blockchain technology with the specific needs of water conservancy projects, and improves the management quality of the electronic files in the concrete mixing process. Meanwhile, this paper also proposes a highly integrated information management system to guarantee the data security of each step in the concrete mixing procedure. The specific steps are as follows: first,

study the relationship between the different concrete production departments and establish a behavioral model describing the concrete production process; second, design and implement a distributed blockchain data structure for concrete production process management; third, use keyless signature technology to manage the type-in, modification, and approval process of the concrete production files; finally, all the files generated in the concrete production process are uploaded to the blockchain to achieve openness and transparency of the entire process, guaranteeing accurate traceability of production files and quick location of quality problems, thus effectively supervising the data quality and safety in the intelligent water conservancy projects construction.

3. Behavioral Model for the Concrete Production Process

3.1. Process Sorting and Role Assignment. To establish a behavioral model for the concrete production process, we first need to sort out the production process. As shown in Figure 1, the concrete production process is divided into raw material preparation and concrete production parts. Specifically, the raw material preparation part can be divided into the import and test subpart. In contrast, the concrete production part can be divided into the mix proportion design subpart, the concrete mixture subpart, and the concrete test subpart.

The raw materials for concrete production include cement, fly ash, admixtures, coarse aggregate, and fine aggregate. The first three materials are transported and supplied by the corresponding manufacturers, while the other materials can be produced by the mixing plant itself. As Figure 1 illustrates, in the material import stage, the quality and quantity reports are provided along with the entry of the purchased raw materials. When the raw materials are in storage, the laboratory of the mixing plant will sample and measure them and then record the report of the material test results in the ledger by computer. Besides, self-made raw materials like coarse and fine aggregates are also tested in detail and recorded by the laboratory of the mixing plant either. At last, all these raw material inspection reports are submitted to the supervision, and the supervision's approval allows the materials to participate in concrete production.

In the concrete production stage, the construction unit submits an application of concrete to the mixing plant. Moreover, the required concrete grade and performance requirements, the required quantity, and the use purpose are also informed to the mixing plant at the same time. After receiving the application, the laboratory personnel in the mixing plant will inspect the moisture content of sand and stone, check the exceeding and inferior grain in aggregate according to the relevant regulations, and then design the concrete mixing proportion. After the supervisor confirms the mixing ratio, the relevant mixing information is provided to the mixing plant. The mixing plant strictly follows the ratio, sets the raw material feeding value, and operates the mixing plant for concrete production. Besides, the raw material temperature and weighing information are recorded during the concrete mixing process according to the regulations. After the concrete mixture, samples are taken from the outlet of the mixing plant; then, the construction unit tests the samples' quality and forms the sample record and test report.

The role assignment could be set as follows by sorting the concrete production process. The main characters involved in the production process are the mixing plant, the laboratory of the mixing plant, the construction department of China Railway 12th Bureau (CR-12 in short), the laboratory of CR-12, the supervisor, and the third-party testing center. In specific, the mixing plant and its laboratory worked in the raw material preparation stage, while CR-12 and the corresponding laboratory worked in the concrete production stage. At last, the supervisor and the third-party testing center took part in every stage of the concrete production process to ensure the safe and reliable quality of the whole concrete production process.

3.2. Classification of Concrete Production Files. The second step of building the concrete production behavioral model is to classify all the files involved in the concrete production process according to their attributes. The files include the raw material performance testing records before concrete mixing, the concrete supply contact sheets, the descriptions on concrete mixing proportion, the records about the mixing process, the result of the concrete performance testing, forms related to each cycle errata, and summaries. The cooperation of these files is demonstrated as follows: The manufacturers supply the raw materials to the mixing plant for concrete production. After production, the mixing plant's laboratory samples the concrete and conducts a quality inspection. If the concrete meets the quality standards, it would be transported to the construction department of CR-12 by vehicles. After the additional tests conducted by the laboratory of CR-12, the construction department of CR-12 builds the water conservancy facilities with qualified concrete. At last, as a neutral third party, the supervisor keeps on inspecting the concrete by commissioning a third-party laboratory to sample and test the concrete at all stages during the production.

In total, after summarizing the files involved in the concrete production process, 50 categories of forms are obtained. There are a total of 29 forms related to raw materials, 1 contact sheet for material supply, 7 forms related to the concrete mixing process, 12 forms related to testing, and 1 form for erratum summary. The details are in Figure 2.

4. Distributed Blockchain Data Structure

4.1. General Framework Design. Based on the behavioral model, the distributed blockchain data structure can be constructed, and then, the preservation, categorization, and management of the concrete production-related archives can be achieved. The general framework design is illustrated in Figure 3. In Figure 3, the archives generated in concrete production are divided into temporal and spatial levels in the order of warehouse blocks, procedure blocks, branch

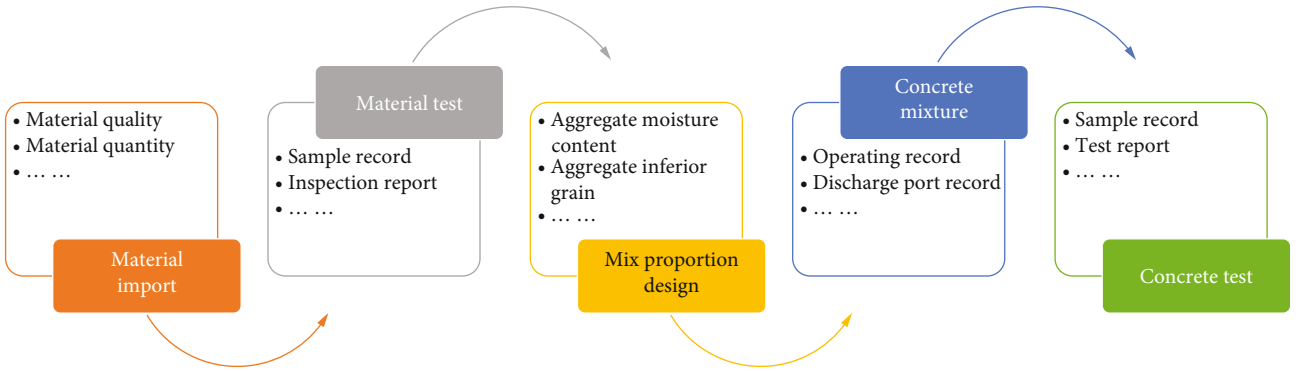

Figure 1: Schematic diagram of the concrete production process.

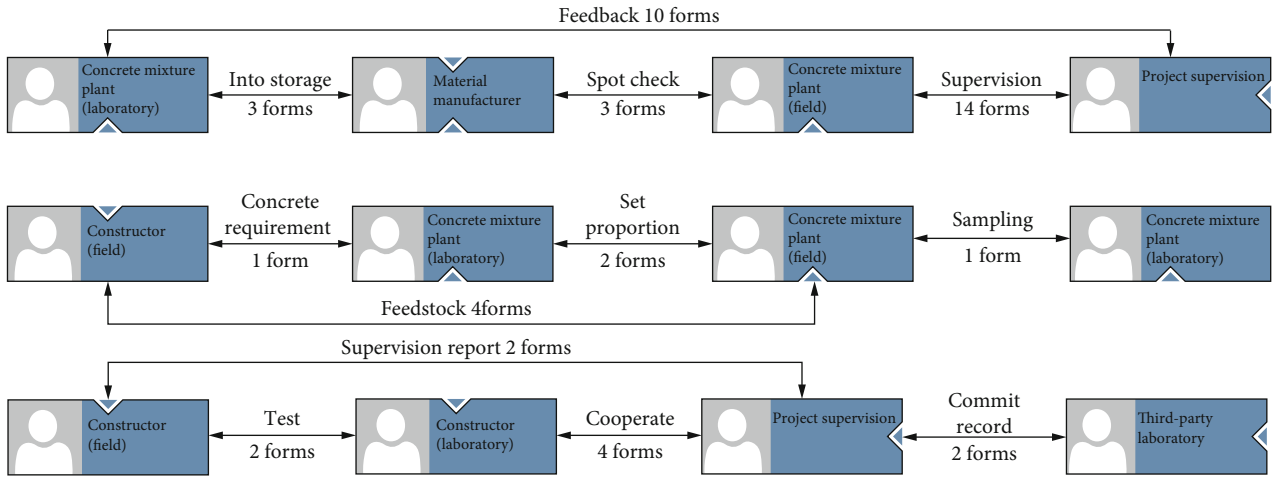

Figure 2: Classification and statistics of the concrete production files.

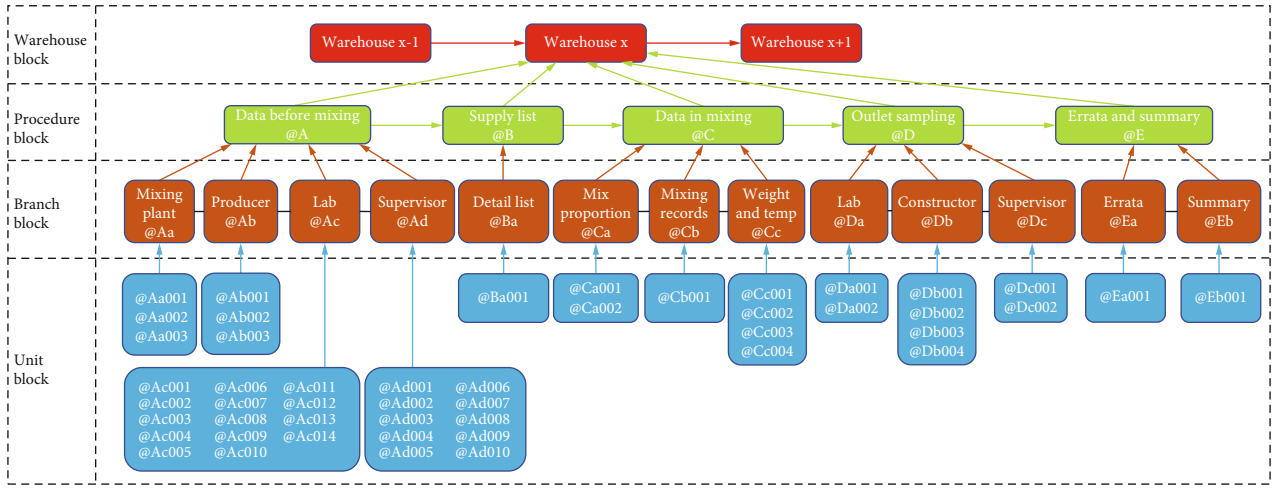

Figure 3: General framework of the distributed blockchain data structure.

blocks, and cell blocks. The structure in Figure 3 represents a comprehensive mixing procedure for a warehouse of concrete, and the warehouse is the fundamental quantity unit in the concrete production process. Inside the data structure, the subblock is composed of one or several distributed ledgers. The functions and properties of each subblock in the distributed blockchain are described below.

The top element in the distributed blockchain data structure is the warehouse block. The warehouse block contains all the files during the concrete mixing process. In the actual

environment, the whole construction procedure of the water conservancy project is often divided into unit projects, division projects, and cell projects. Furtherly, the cell projects are refined into a series of sequential subtasks, and the data and corresponding files generated in each subtask are formed as a warehouse block. Like the example in Figure 3, during the mixing process, the computers automatically record the production data for each tray of concrete, including the set and actual usage amount of the raw materials, the mixing time, the use of the concrete, and other detailed information. The warehouse blocks are made up of cyclic packets, and they are numbered sequentially from 0001 onwards in chronological order.

The procedure blocks are the blocks that indicate the specific flows of the concrete production. Note that the block could not be formed until the previous one is generated, and all blocks in the same layer are chained together in a tandem pattern. As shown in Figure 3, the concrete production process contains five procedure blocks: data before mixing, supply list, data in mixing, outlet sampling, errata, and summary.

The blocks in the branch block layer are the distributed ledgers created and categorized by different roles in the concrete mixing procedure. For example, @Aa,@Ab,@Ac,@Ad are the branch blocks under the same procedure block A in Figure 3; they represent the file collections in the mixing plant, the producer, the lab, and the supervisor, respectively.

The bottom layer in the distributed blockchain data structure is the unit block layer. In this layer, the unit blocks are the specific files, forms, images, or other media boundaries with archival requirements in branch blocks, marked with 001, 002, and so on. As shown in Figure 3, each unit block refers to one file created by a specific role.

Besides, the naming scheme of the proposed distributed blockchain data structure is as follows: Firstly, "@" and "#" in the front of each unit block number indicate if this block is shared or not. Secondly, the warehouse block is often divided into blocks for the raw material test, blocks for the concrete inspection, and the other blocks. Among them, the raw material inspection blocks record the samples and the test results of raw materials, such as 200~400 t a sampling unit of cement, 100~200 t a sampling unit of fly ash, and 50 t a sampling unit of admixture. Thirdly, if the unit block is shared, the provenance of the shared data should be indicated, and the indication method is to add the name of the warehouse block which contains the shared block. For instance, if the cement test report is "×××××Ac013," suppose that a new cement test report is generated in warehouse block "0020," then its number is "0020#Ac013." If the next five bunker blocks "0021," "0022," "0023," "0024," and "0025" need to quote the previous report rather than generating new cement inspection reports, then the quoted report is named as "0020@Ac013." But if the warehouse block "0026" generates a new cement inspection report, then the name of the report is "0026#Ac013."

4.2. Distributed Storage Architecture for Digital Archives. Based on the design of blockchain data structure, this paper classifies the files according to different production roles and then stores them in distribution. Specifically, the main characters participating in the concrete production procedure keep their own files locally. For instance, the construction department that initialized and transmitted the supply list would leave a copy of the list in the server of CR-12. Similarly, if the mixing plant initiates the batching notification form, then the form is stored in the computer of the mixing plant. The rules for the rest of the file storage locations are similar, except that the files that are shared by different branches should be stored by both the sending and the receiving units.

Moreover, the data-sharing scheme is another crucial part of distributed storage architecture, and it consists of two parts: the data sharing between files and inside files. The data sharing between files means keeping the same sections' consistency and accuracy in different files. There is a mapping or logical relationship between information in some files and information in the other files during transmission. Therefore, when creating such files, we first store this information in public memory and then automatically obtain the corresponding data with the same content on different files. For illustration, "construction site, strength grade, collapse level, and planned quantity" in the batching order are derived from the contents "construction site, seepage and frost resistance, collapse level and outlet temperature, and concrete supply order" in the supply list. Similarly, the "oversize content" and "undersize content" on the batching notice come from the same contents on the coarse aggregate test records. However, if the shared data is inconsistent, we will issue warning messages to senders and receivers. Then, the file is rejected by the receiver until the sender makes corrections. During the revision, the character who makes the file first checks if the inconsistency is indeed caused by himself then resolves this dispute by amending the filled-in content. Otherwise, the inconsistency is caused by the incorrect data in the system. Then, the dispute will be temporarily put on hold through the consensus mechanism and resolved through the errata at the end of this warehouse block.

The data-sharing scheme inside files means that the files are shared between different warehouse blocks. For illustration, in the raw material test stage, an inspection form for raw materials may cover more than one warehouse block; then, these blocks share the same inspection form. As mentioned above, the specific method to distinguish the shared and the unshared data uses "@" and "#" symbols as indicators.

5. Edge Computing Supported Intelligent Keyless Signature

During the concrete mixing, every authentic and valid file requires the principal's signature of every department, and the signature means the approval of the file content. This signature process is represented as the form-filling operation in the proposed model. However, there is a risk of tampering with the file during the filling process. The traditional approach is to introduce asymmetric encryption technology in the file approval process to ensure the security of

transmission and the file's integrity. But this technical approach has certain management risks because it involves the management of an individual's private key. Therefore, this paper employs a keyless signature technology based on edge computing to standardize the form-filling process and provide security for electronic files.

5.1. Hash Tree Construction Based on Edge Computing. The fundamental method for data security during the file transmission is to use the hash function to make a calculation on the file and then regard the calculation result as a digital fingerprint to prove the file's authenticity. In detail, the proposed management model uses the SHA-256 hash algorithm to calculate the file, generate a 256-bit hash value, perform a series of operations with the hash value, and build up a hash tree. The process is in Figure 4.

In Figure 4, x_1 to x_8 represent the hash values calculated with the SHA-256 algorithm, and these values are the input of the leaf nodes in the hash tree. $h()$ denotes the hash function, and the vertical line represents the join operation, but $h(x_1 \mid x_2) \neq h(x_2 \mid x_1)$. The hash tree introduces the hash function to fulfill zero-knowledge proof and ensure that the file is authentic. For example, suppose the initial data x_3 knows the hash values $\{x_4, x_{12}, x_{58}\}$ and their position markers $\{1,0,1\}$. In that case, the root value can be recreated, thus proving that x_3 is involved in calculating the generated root value. In total, based on hash chains, goals including a fast comparison on massive data, locating the modified data, and constructing zero-knowledge proofs, can be easily achieved. The hash chain computing process is shown in Figure 5.

Further, to secure data transfer and file integrity from the spatial dimension, a large number of hash trees need to be aggregated into Merkle trees simultaneously, and edge computing is the best way to achieve such goals. A Merkle tree consists of a root node, some intermediate nodes, and a set of leaf nodes. Each leaf node is labeled with the hash value of the digital file, while intermediate nodes other than the leaf nodes are marked with the cryptographic hash of their child node labels. Creating a complete Merkle tree requires recursively hashing a set of nodes and inserting the generated hash nodes into the tree until only one hash node remains, which is also called the Merkle root. The construction process of the Merkle tree is in Figure 6.

As shown in Figure 6, Merkle trees are created and destroyed once per second. These trees are composed of a hierarchical network of geographically independent distributed computing nodes. Each operates in an asynchronous aggregation fashion, generating a hash tree by receiving hash values from its subtrees transmitting the hash root values to multiple parents. The aggregation process is theoretically unbounded and runs on top of virtual machines or dedicated hardware. Moreover, in a keyless signature system with a multilayer aggregation hierarchy, the acceptable theoretical limit of the system is 2^{64} signatures per second.

5.2. The Intelligent Keyless Signature System. The keyless signature system based on Merkle trees is shown in Figure 7, and the specific tree construction process can be described as follows. Firstly, the department participating in the concrete mixing procedure submits the hash value (the blue dots in Figure 7) of the file to the customized keyless signature gateway. Secondly, the adjacent hash values are connected in series, and then, an additional hash operation on the concatenated values is performed again to calculate the result. Subsequently, the newly calculated hash value is submitted to the upper layer for serial hash operation until the Merkle tree's root is created. Finally, the keyless signature gateway returns a keyless signature to the department. The keyless signature contains the hash value submitted in the previous step and the sequence to regenerate the hash root value. This keyless signature is a hash chain composed of coordinates like the red dots in Figure 7. With this keyless signature system, the concrete construction department can ensure the spatial integrity of electronic data.

Except for guaranteeing the spatial integrity of the electronic files, the intelligent keyless signature system based on Merkle trees can also ensure the temporal reliability of the electronic files. The mechanism is illustrated as follows: First, the keyless signature system stores the hash root values in a shared database called the calendar database while creating and destroying every second. Specifically, since 0 : 00, 0 seconds on January 1, 1970, each second of hash values has been regarded as a leaf node, forming a particular type of permanent hash tree, also known as a Merkle forest. The calendar hashes are periodically aggregated to generate the integrity code's hash value. In a keyless signature system, the calendar database's integrity code is regularly issued in electronic and paper form in the world media, as shown in Figure 8 [27]. After the integrity code is released in the electronic or paper-based public media, the authenticity of all signatures can be evaluated by tracing back the integrity code, thus ensuring the temporal integrity of the data. [28].

5.3. Signing and Verification of Production Files. Signing and verifying the production files based on keyless signature are illustrated in Figure 9. As the description at the top of Figure 9, when a file is created and needs to be signed during the concrete production process, first, the signatories make a hash calculation on the file with the SHA-256 function and then submit the hash value to the distributed keyless signature server. From the one-way nature of the hash function, it is clear that the hash value is only the credential for applying a keyless signature, so the privacy of the original file is still kept. In the second step, the keyless signature server that receives the hash value performs a calculation through the hash chain and returns a keyless signature starting from the root node of the Merkle tree to the signatories as a response. In the third step, the keyless signature server timely releases the integrity code through newspapers or other forms. Note that the integrity code is preserved in the online calendar database after its release.

The verification of signed files usually occurs in the file approval stage. As shown at the bottom of Figure 9, when the validator receives a signed file from the previous signatory, in order to verify the authenticity of the data, first and foremost, the received file and its corresponding keyless signature should be aggregated to conduct a hash

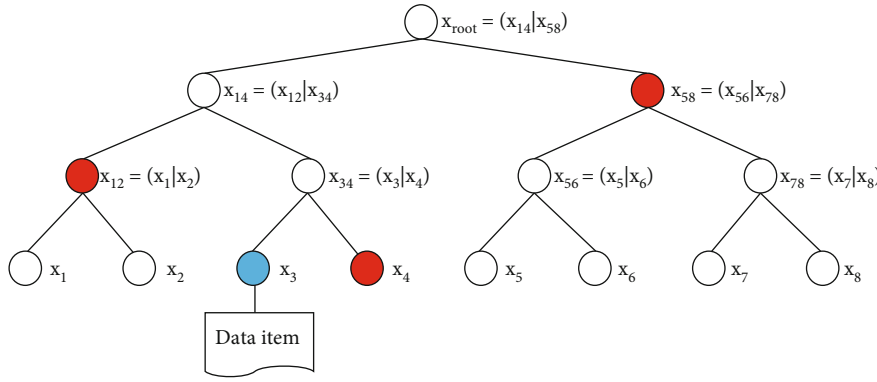

Figure 4: Schematic diagram on hash tree construction.

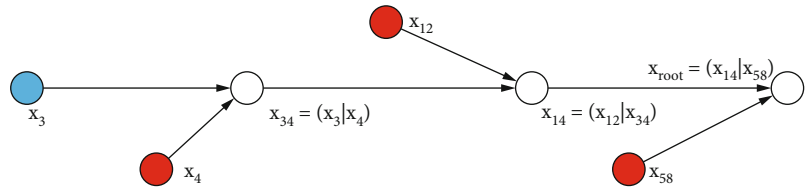

Figure 5: Schematic diagram on hash chain computing.

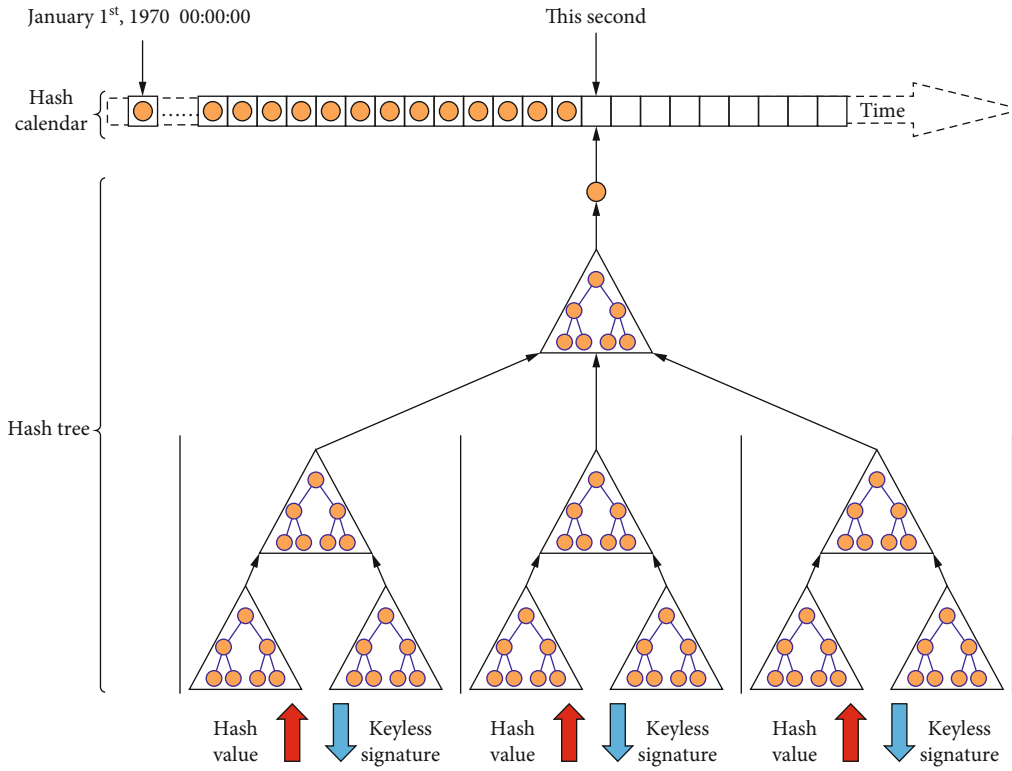

Figure 6: Parallel construction of the Merkle tree based on edge computing.

computation. Next, the integrity code associated with the signature file is figured out from the online database. Then, a comparison of the integrity code with the hash computation result is conducted subsequently. If the comparison result is consistent, it indicates that the signature data is accurate and trustworthy, the file transmission and approval process is in line with the standard requirements, and there is no tampering with the data. If the comparison result is

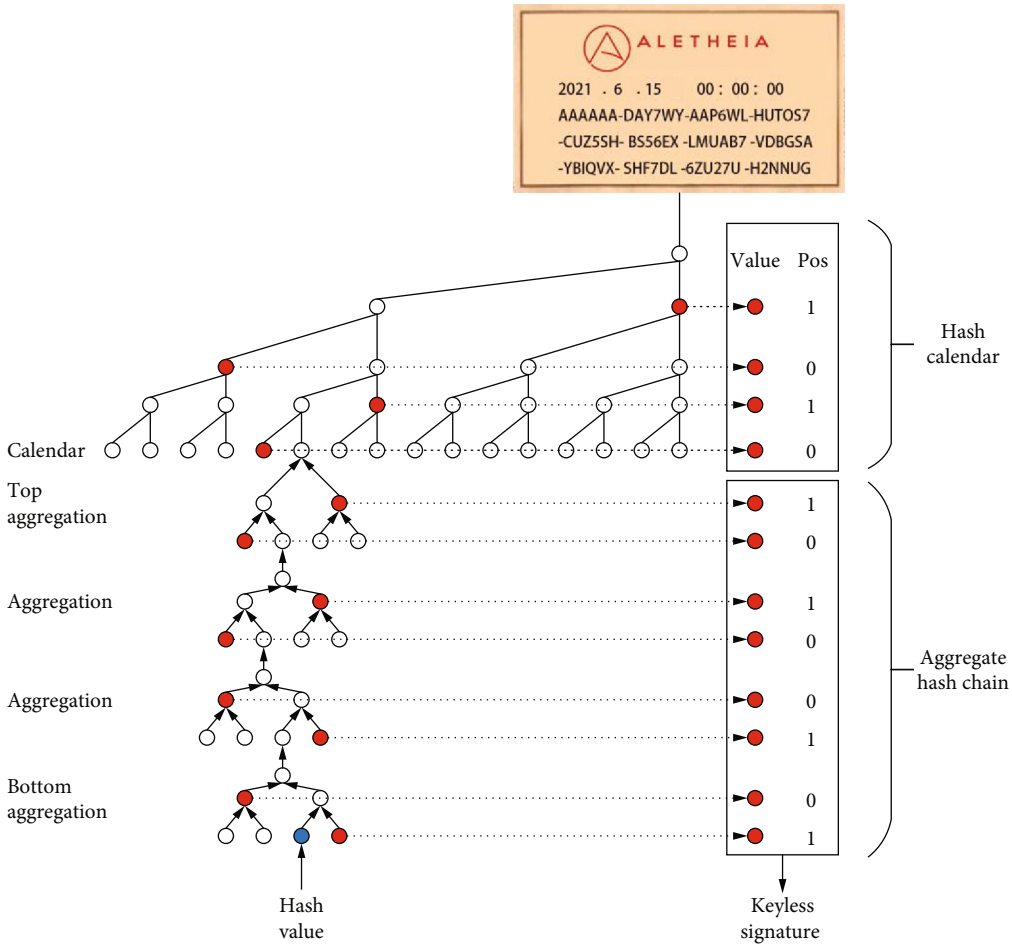

Figure 7: The keyless signature system legend.

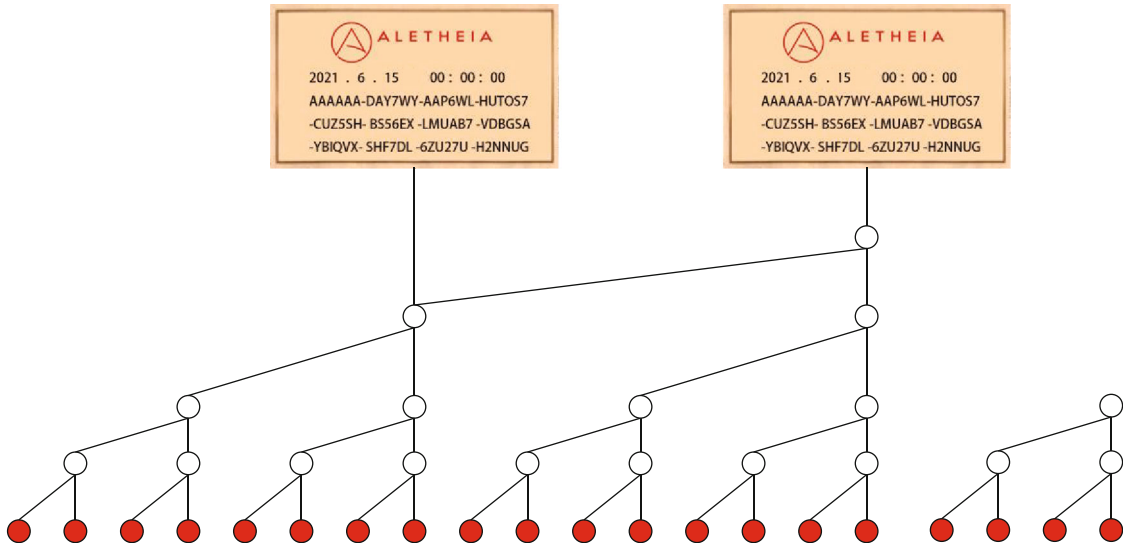

Figure 8: The Merkel forest structure.

inconsistent, it proves that the concrete production file management process is not standardized and data security risks have occurred.

In summary, the implementation of the keyless signature system could standardize the approval and writing process of files during the concrete production process, supervise

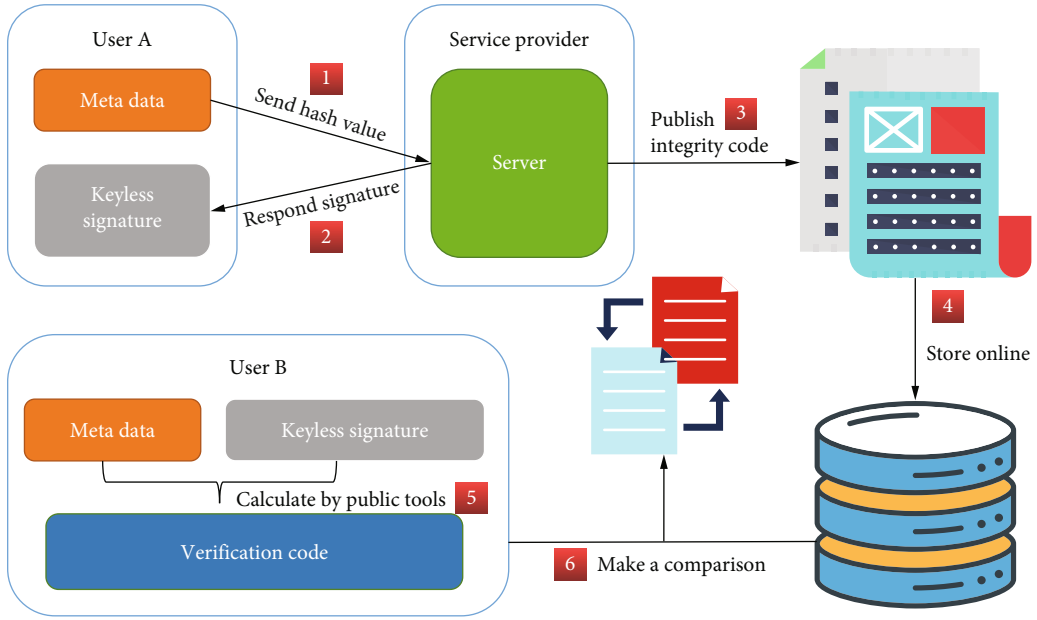

Figure 9: Data signing and verification process based on the keyless signature.

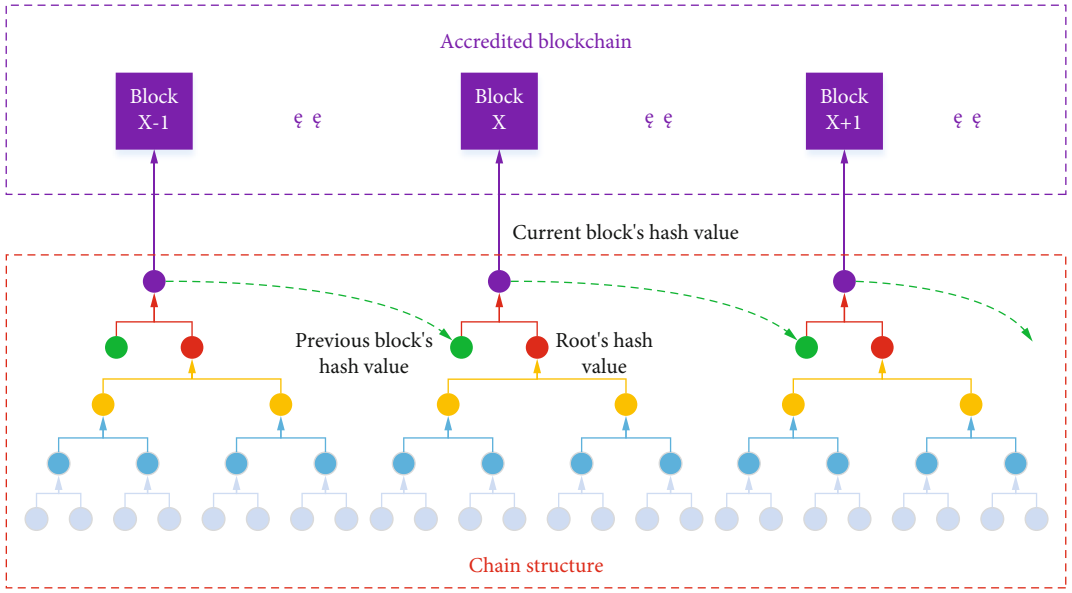

Figure 10: Schematic diagram of the chain structure.

every step of data generation, and eliminate irregular data recording and internal tampering, thus protecting the security of the concrete production file to a higher degree and for a long time.

6. Production File Management Based on Blockchains

6.1. Chain Structure Design. Based on the data structure and the keyless signature system, the chain structure of the concrete production file and the corresponding data on-chaining process are in Figure 10. When all the files involved in each concrete warehouse are collected, each file's hash value, also known as the unit block in the distributed blockchain data structure, is calculated separately. Then, a series of unit blocks aggregate two by two to form a binary tree, the root of which is called a procedure block. Thirdly, many procedure blocks polymerize to a compound as a Merkle tree, and the root is regarded as the warehouse block. Finally, by aggregating the current warehouse block with the previous warehouse block into a Merkle tree and storing the root of the tree on a trusted blockchain, the information security of the adjacent two warehouse blocks can be ensured.

As Figure 10 illustrates, compared with the traditional paper form files, the electronic files are more conducive to data search and analysis. Besides, electronic information

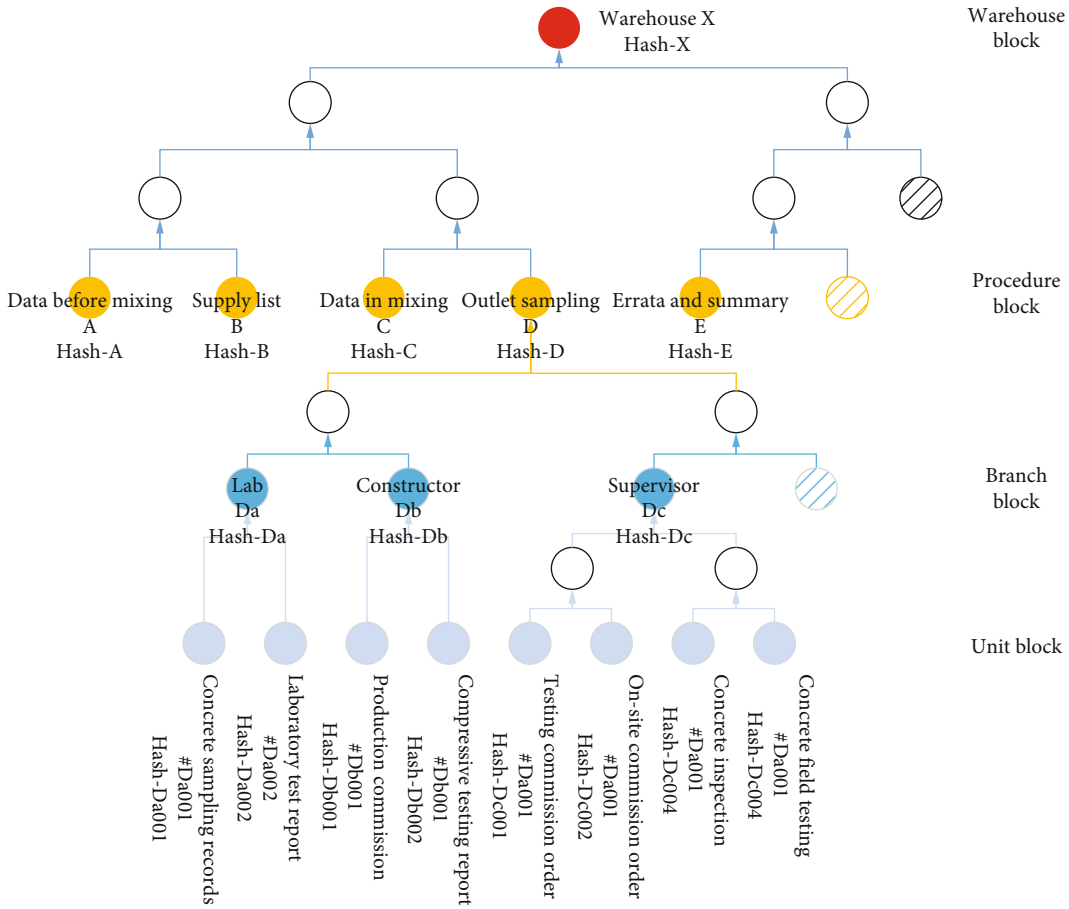

Figure 11: The organization of the warehouse block.

security and traceability can be improved markedly by the blockchain-based information deposition mechanism compared with the conventional centralized storage database.

6.2. Automatic Data On-Chaining Mechanism. Creating one warehouse block and uploading it onto blockchain means that the mixing plant finishes an entire concrete production task from batching, mixing to the end according to the instructions. A warehouse usually produces tens to hundreds of cubic meters of concrete. In the proposed model, the warehouse blocks are at the top level, and adjacent warehouse blocks are linked in tandem with time stamps. The organization of each warehouse is in Figure 11.

As shown in Figure 11, the blocks of each warehouse employ the Merkle tree structure to organize data, which is compatible with the signature generation mechanism in the keyless signature system. In Merkle trees, the two leaf nodes on each set of forks represent two files, and the files are paired two-by-two in the order of their generation time. In Figure 11, the file hash is regarded as a unit hash, and the branch hash is generated by two-by-two aggregation of all unit hashes. Furtherly, the procedure hash is composed of a two-by-two accumulation of branch hashes, and the warehouse hash is made up of pair-wise procedure hashes. Finally, the automatic data uploading is finished when all unit hashes are chained to form a warehouse hash.

Due to the structural characteristics of the Merkle tree, any changes in the underlying data will lead to changes in its parent nodes and eventually affect the changes in the Merkle root. So the Merkle tree has the advantages of efficient comparison of a large amount of data, fast location of modified data, and fast verification of incorrect data, which are all demonstrated explicitly in the proposed management model. For illustration, when two Merkle tree roots are the same, the data they represent must be the same, which makes data verification between different users possible. Besides, when the underlying data is changed, its location can be quickly detected by inspecting the corresponding branch. With this feature, the proposed model can easily fulfill fast querying of the information about the abnormal data. Last but not least, when it is necessary to prove the originality and authenticity of the data, only the hash summary of the data needs to be validated without knowing the exact content of the data.

6.3. Smart Contracts and Consensus Algorithm. The smart contracts in our blockchain management model are fulfilled by introducing various forms of notification measures such as emails and cell phone applets to inform users of pending matters and remind them of the approval delays during file flow. Beyond that, to ensure data consistency during the automatic data on-chaining process, our model adopts

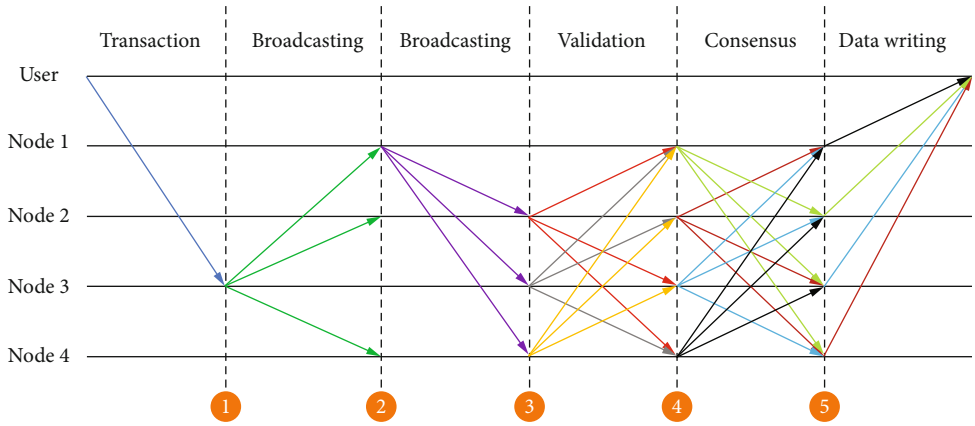

Figure 12: The implementation model of the consensus mechanism.

Byzantine Fault Tolerance (BFT) [29] as the consensus algorithm to synchronize the data to be recorded. The automatic concrete production data on-chaining mechanism based on the BFT consensus algorithm is shown in Figure 12, and it can be precisely divided into five steps, which are as follows:

(1) When the supervisor starts the approval of the concrete mixing order, the action will be considered a transaction, and the proposed model broadcasts this transaction to all blockchain nodes, including raw material providers and construction units

(2) After hash computation, the supervisor broadcasts the hash value of the transaction to all blockchain nodes

(3) Each blockchain node (participating construction unit) makes a hash after receiving the transaction and compares it with the supervisor's hash sequence

(4) After all nodes receive the message that more than half of the comparisons are approved, the transaction is deemed to be established

(5) The transaction is recorded into the block

6.4. Validation and Abnormal Block Tracking. The validation and tracking of the production files can also be fulfilled by the blockchain. As Figure 13 shows, the process is to verify the file's integrity and record the location of the files that failed the verification. Specifically, the information of the abnormal file is retrieved from the database; then, the values of the file and the corresponding warehouse are recalculated according to the calculation rules at the time of uploading. Subsequently, the newly calculated warehouse values are compared with the corresponding uploaded warehouse values on the blockchain in sequence according to the warehouse organization order.

Suppose the comparison of the hash values is consistent. In that case, all the electronic files in the warehouse are safe and secure. It has not been tampered with, so it is unnecessary to continue comparing the detailed information of this warehouse. But if inconsistency happens, the files contained in that warehouse are lost or tampered with, so it is necessary to continue to compare the hash value of each file in that warehouse. The processes of file hash matching and warehouse hash matching are the same; the newly calculated file hash is compared with the file hash recorded on the blockchain. The file that contains inconsistent hash comparison results is recorded. Thus, the traceability of the problematic blocks can be achieved.

7. Model Application

7.1. Overall Architecture. In this paper, a concrete production management system based on the proposed model has been developed and implemented in the Hanjiang to Weihe River Project in Shaanxi Province to verify the model's practicality and security. The system adopts a B-S architecture, and all users can log in and use it directly through a browser. Figure 14 shows the overall architecture.

The concrete production information management system mainly manages data related to concrete production in the water conservancy project construction, including standardized management of file filling, unified management of data archiving, and automatic uploading of production files. The management system consists of user management, menu management, process management, parameter management, authority management, and log management. Through the network interface provided by the management system, different construction units in the concrete production system automatically import or manually enter various information about concrete production and create electronic files. After that, the file, branch, procedure, and warehouse hash are generated sequentially, and then, they are organized to the tree structure according to the distributed blockchain data structure.

The generated hash values are uploaded to a credible blockchain for deposition. The information interaction between the blockchain and the information management platforms is fulfilled through port calls. In our information management system, the blockchain is the consortium blockchain called the Blockchain-based Service Network. This blockchain was jointly initiated by the State Information Center, China Mobile Communications Corporation, China UnionPay Corporation, and Beijing Red Date

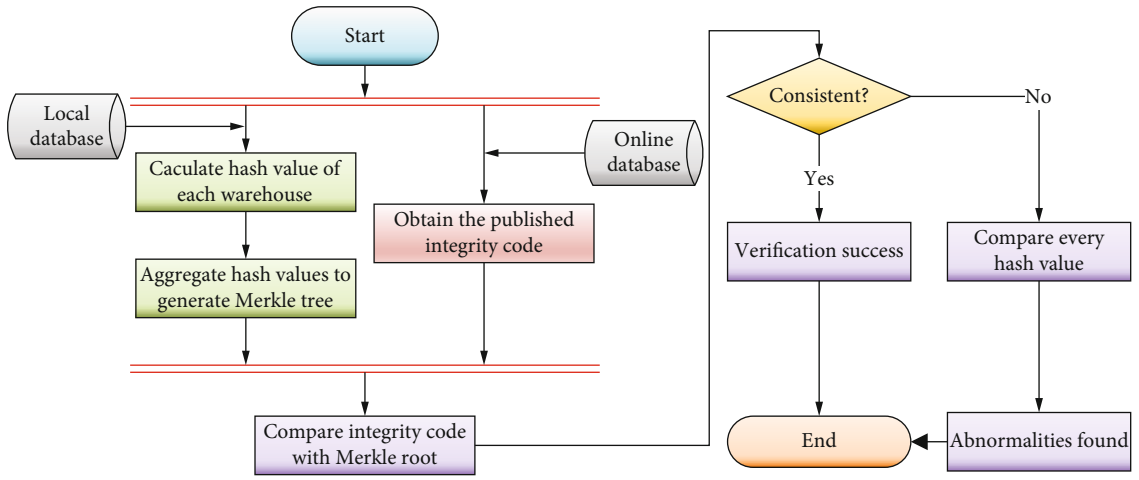

Figure 13: Schematic diagram of the abnormal block tracking flow.

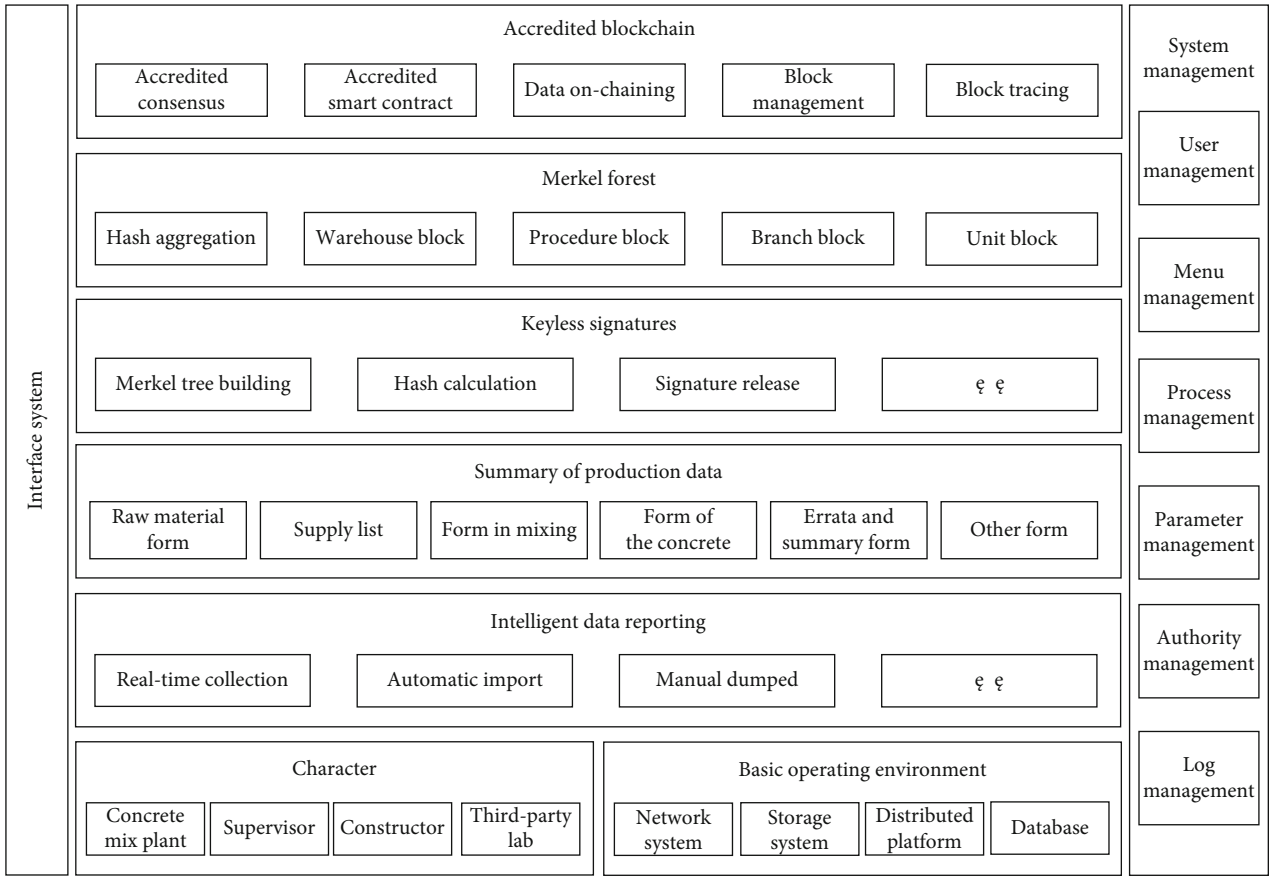

Figure 14: Architecture of concrete production information management system.

Technology Corporation [30]. Besides, this consortium blockchain provides the storage, verification, and traceability of hash values and facilitates historical data security verification.

In practice, the system was implemented in the Hanjiang to Weihe River Project to collect and organize the concrete production-related files in 2020. The total concrete production volume in the project in 2020 was about 170,000 square meters, which generated about 16,000 related paper forms in total. At present, we have entered and uploaded some of the files, including 18,000 square meters of concrete related to more than 3,500 forms, and stored these records on the consortium blockchain. The data server and the application server configurations in our system are the same: both are

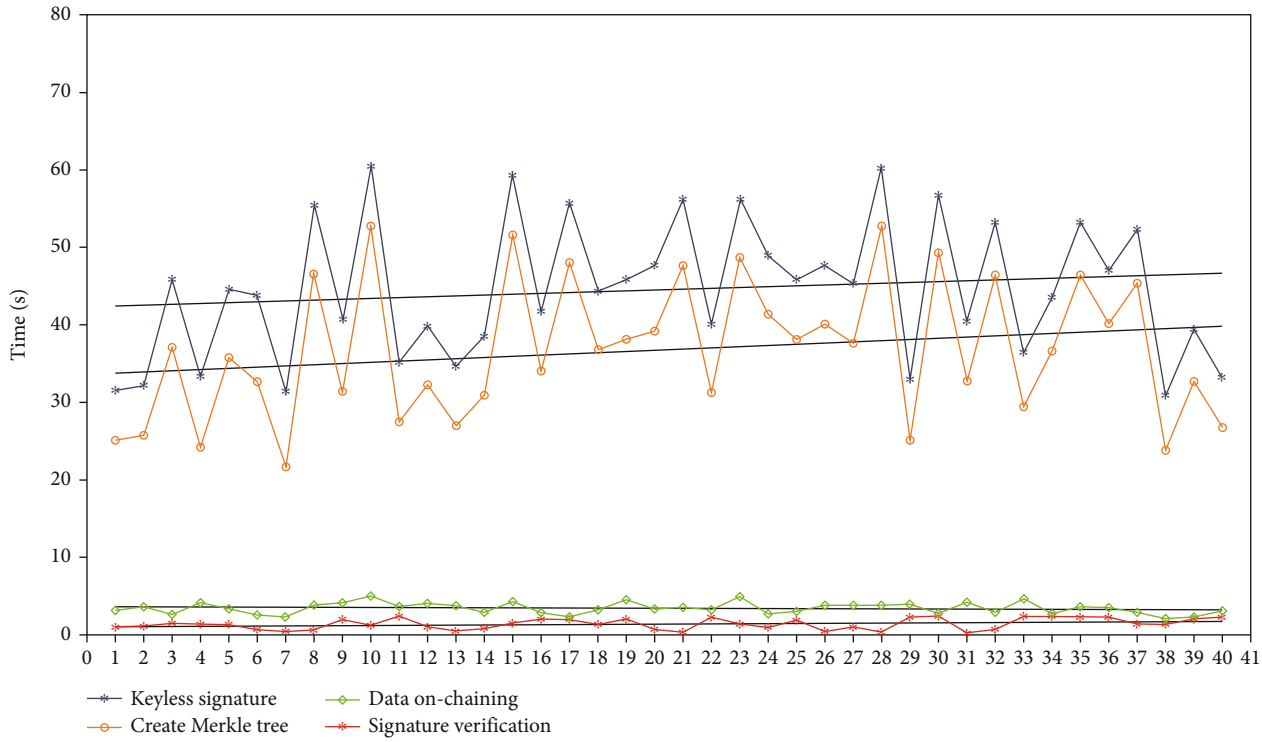

Figure 15: Comparison of system time consumption by phase.

dual-core and quad-threaded, with 8 G memory and 500 G storage space, which meet the minimum requirements for civil engineers [31].

7.2. Experiment and Analysis. The experiment for testing the system performance is designed as follows: one warehouse block is selected randomly as the experiment object from the actual concrete production process. The target warehouse block contains 40 files, including supply contact sheets, production notification sheets, quality inspection sheets, and errata summary sheets, recording a complete concrete production process. The file creation and transmission are standardized with the keyless signature. After the files are filled and verified, they are uploaded on the blockchain for permanent storage. Besides, the time for the Merkle tree construction, the keyless signature creation, the signature verification, and files' chaining are recorded separately. The specific time spent on the four steps of the 40 concrete production files is shown in Figure 15.

As shown in Figure 15, the overall trend of keyless signature generation time per file raises as the production data increases. By fitting the linear regression model, it can be seen that the slope of the total keyless signature time is about 0.109. For each integrated keyless signature registration, when the size of the Merkle tree increases, the keyless signature generation time of the following file will also increase by about 0.109 s.

Secondly, the average time consumption for data on-chaining is about 3.39 s, with a slope of -0.009. This erratic fluctuation is caused by blockchain instability and network fluctuations.

Thirdly, the time for the Merkle tree generation also shows an increasing trend correlation with the keyless signature generation time because the keyless signature is based on the combination of hash values from the root to the leaf sequence of the Merkle tree and its corresponding sequence coordinates. The slopes of Merkle tree creation and keyless signature generation are similar by linear fitting, which indicates that the creation time of the Merkle tree is the main factor that increases the generation time of keyless signature.

Fourthly, the average time to verify the on-chain data is about 1.38 s. The slope of the linear fit function is 0.019, indicating that the verification is swift for on-chain data. The verification efficiency is mainly affected by the structure of the warehouse block.

Besides, we also conduct the tests on keyless signature sizes. As shown in Figure 16, the keyless signature size of each file is about 157 kb, and its storage cost is less than 1 penny. The keyless signature storage cost of the whole warehouse is less than 0.1 yuan, and this cost is almost negligible compared with the benefits of data security.

Finally, we use the number of transactions processed per second as the criterion for system throughput to evaluate the entire performance. The throughput of the relevant smart contracts is calculated for different concurrent requests. The number of concurrent requests is set from 100 to 1000, and 10 experiments are conducted in sequence. At last, the average values are taken as the experimental results. The throughput of the smart contracts is in Figure 17.

In Figure 17, the throughput of the write operation (data on-chaining) is overall lower than that of the read operation (signature verification). In other words, the write operation

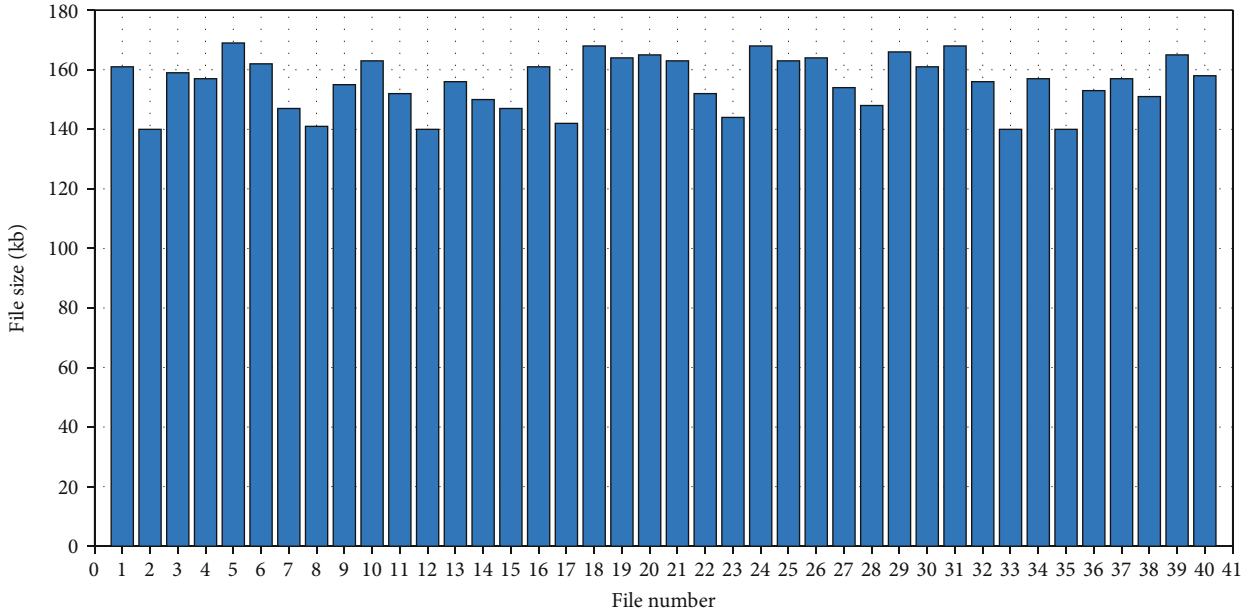

Figure 16: Storage space consumed by keyless signatures.

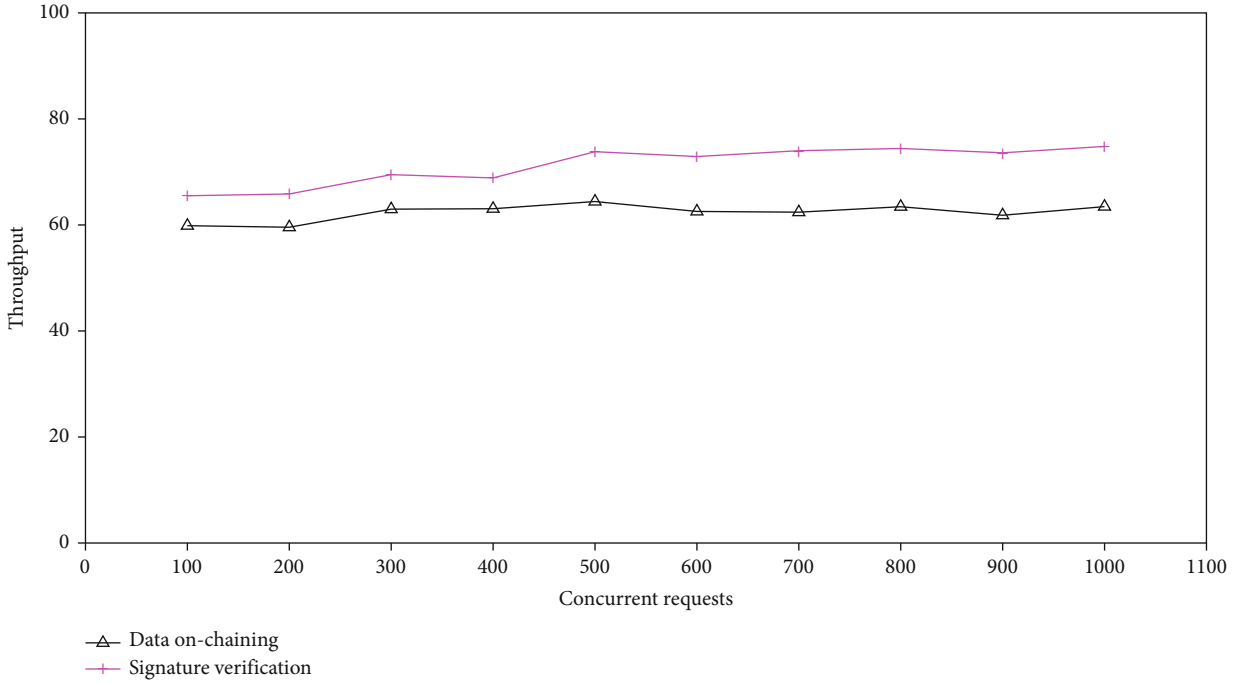

Figure 17: Throughput of the smart contract with the concurrent requests.

tends to be more time-consuming than the read operation. It is because the write operation needs to hash the data and generate a historical version of the old data to ensure the traceability of the blockchain. On the other hand, read operation only needs to search and validate data based on index positions, thus taking less time. Besides, the system's throughput increases with the number of concurrent requests. However, when the number of simultaneous requests reaches a certain value, the growth trend slows down slightly. By calculation, the throughput stays 71 for the smart contract with data on-chaining. In contrast, the throughput for a signature verification smart contract is about 62.

8. Conclusions

The digital archiving of engineering construction files in intelligent water projects is of great significance. The blockchain can provide security verification and integrity check for electronic files, which guarantees the security of archive

informatization and contributes to the realization of electronic archiving of files. This paper proposes a comprehensive data management model for smart water system construction based on blockchain and edge intelligence and then implements it in the Hanjiang to Weihe River Project in Shaanxi Province. Firstly, the behavioral model for the concrete production process is summarized, and the corresponding roles that participate in the process are abstracted out simultaneously. Secondly, the intelligent keyless signature based on parallel edge computing is introduced to ensure data security. The proposed model uses the Merkle tree to construct a chained file structure and standardizes the data entering, uploading, and checking procedure by the consensus mechanism. In the case study, we have created a blockchain of 3,500 blocks according to the decentralization requirement. In total, the proposed model and the corresponding system have already taken a big step forward in saving workforce and material resources and improving the security and traceability of construction archives markedly. We believe that through a more extensive scope of application and continuous improvement, the management of archives in civil engineering, especially in smart water projects, will eventually achieve the goal of digitalization.

Data Availability

The experiment data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

All authors declare no conflict of interest in this paper.

Acknowledgments

This research work is supported by the National Natural Science Funds of China (62072368), Basic Research in Natural Science and Enterprise Joint Fund of Shaanxi (2021JLM-58), and Special Scientific Research Project of Education Department of Shaanxi (21JK0781).

References

[1] N. Nizamuddin, K. Salah, and M. A. Azad, "Decentralized document version control using ethereum blockchain and IPFS," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[2] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of drones," *IEEE Internet of Things Journal*, 2021.

[3] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[4] Y. Zhang, W. Luo, and F. Yu, "Construction of chinese smart water conservancy platform based on the blockchain: technology integration and innovation application," *Sustainability*, vol. 12, no. 20, p. 8306, 2020.

[5] J. Song, Z. Han, W. Wang, J. Chen, and Y. Liu, "A new secure arrangement for privacy-preserving data collection," *Computer Standards & Interfaces*, vol. 80, article 103582, 2022.

[6] C. Feng, B. Liu, and K. Yu, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, 2022.

[7] T. Bui, D. Cooper, and J. Collomosse, "Tamper-proofing video with hierarchical attention autoencoder hashing on blockchain," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2858–2872, 2020.

[8] B. Zhong, H. Wu, and L. Ding, "Hyperledger fabric-based consortium blockchain for construction quality information management," *Frontiers of Engineering Management*, vol. 7, no. 4, pp. 512–527, 2020.

[9] L. Zhang, M. Peng, and W. Wang, "Secure and efficient data storage and sharing scheme for blockchain-based mobile-edge computing," *Transactions on Emerging Telecommunications Technologies*, no. article e4315, 2021.

[10] G. Nagasubramanian, R. K. Sakthivel, and R. Patan, "Securing e-health records using keyless signature infrastructure blockchain technology in the cloud," *Neural Computing and Applications*, vol. 32, no. 3, pp. 639–647, 2020.

[11] J. Zhang, Y. Huang, and Y. Wang, "Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms," *Construction and Building Materials*, vol. 253, article 119208, 2020.

[12] Y. Gong, L. Zhang, and R. Liu, "Nonlinear MIMO for industrial Internet of Things in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5533–5541, 2021.

[13] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[14] K. A. Nguyen, R. A. Stewart, H. Zhang, O. Sahin, and N. Siriwardene, "Re-engineering traditional urban water management practices with smart metering and informatics," *Environmental Modelling & Software*, vol. 101, pp. 256–267, 2018.

[15] J. Feng, L. Liu, and Q. Pei, "Min-Max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, 2022.

[16] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the Internet of health things," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, 2021.

[17] K. Yu, M. Arifuzzaman, and Z. Wen, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE transactions on instrumentation and measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.

[18] L. Tan, K. Yu, and N. Shi, "Towards secure and privacy-preserving data sharing for covid-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, 2022.

[19] L. Zhao, J. Li, and A. Al-Dubai, "Routing schemes in software-defined vehicular networks: design, open issues and challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 4, pp. 217–226, 2020.

[20] J. Feng, F. R. Yu, and Q. Pei, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2019.

[21] L. Tan, K. Yu, F. Ming, X. Chen, and G. Srivastava, "Secure and resilient artificial intelligence of things: a HoneyNet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.

[22] K. Yu, L. Tan, and L. Lin, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote E-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[23] L. Liu, C. Chen, and Q. Pei, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[24] S. Hakak, W. Z. Khan, and G. A. Gilkar, "Securing smart cities through blockchain technology: architecture, requirements, and challenges," *IEEE Network*, vol. 34, no. 1, pp. 8–14, 2020.

[25] K. Yu, Z. Guo, and Y. Shen, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, 2022.

[26] T. Alladi, V. Chamola, and R. M. Parizi, "Blockchain applications for industry 4.0 and industrial IoT: a review," *IEEE Access*, vol. 7, pp. 176935–176951, 2019.

[27] H. Huang, J. Lin, and B. Zheng, "When blockchain meets distributed file systems: an overview, challenges, and open issues," *IEEE Access*, vol. 8, pp. 50574–50586, 2020.

[28] L. Liu, J. Feng, and Q. Pei, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2020.

[29] H. Xiong, C. Jin, M. Alazab et al., "On the design of blockchain-based ECDSA with fault-tolerant batch verication protocol for blockchain-enabled IoMT," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, p. 1, 2021.

[30] J. Ma, S. Zhang, and H. Li, "Sparse Bayesian learning for the time-varying massive MIMO channels: acquisition and tracking," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 1925–1938, 2018.

[31] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.