Received October 31, 2019, accepted November 6, 2019, date of publication November 11, 2019,
date of current version November 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952635

Blockchain Based Data Integrity Verification for Large-Scale IoT Data

HAIYAN WANG AND JIAWEI ZHANG

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 21003, China
Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China
Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210023, China

Corresponding author: Haiyan Wang (wanghy@njupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772285, and in part by the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing.

ABSTRACT Achieving data integrity verification for large-scale IoT data in cloud storage safely and efficiently has become one of the hot topics with further applications of Internet of Things. Traditional data integrity verification methods generally use encryption techniques to protect data in the cloud, relying on trusted Third Party Auditors (TPAs). Blockchain based data integrity schemes can successfully avoid the trust problem of TPAs, however, they have to face the problems of large computational and communication overhead. To address the issues above, we propose a Blockchain and Bilinear mapping based Data Integrity Scheme (BB-DIS) for large-scale IoT data. In our BB-DIS, IoT data is sliced into shards and homomorphic verifiable tags (HVTs) are generated for sampling verification. Data integrity can be achieved according to the characteristics of bilinear mapping in the form of blockchain transactions. Performance analysis of BB-DIS including feasibility, security, dynamicity and complexity is also discussed in detail. A prototype system of BB-DIS is then presented to illustrate how to implement our verification scheme. Experimental results based on Hyperledger Fabric demonstrate that the proposed verification scheme significantly improves the efficiency of integrity verification for large-scale IoT data with no need of TPAs.

INDEX TERMS Data integrity verification, blockchain technology, bilinear mapping, Internet of Things (IoT).

I. INTRODUCTION

With the wide popularity of Internet of Things (IoT) technologies such as smart cities, autonomous vehicles and smart grids, the number of devices connected to the Internet is rising overwhelmingly. According to Gartner's forecasts, there will be a 42% increase in IoT connections and $20 billion in spend from 2018 to 2020. How to collect [1], process, store and analyze these large-scale IoT data securely [2] has therefore become one of the most important issues for further applications of Internet of Things. Traditional distributed database systems cannot satisfy the requirements of data management in the IoT environment, and Cloud Storage Services (CSSs) arise consequently.

With external storage of data, the integration of IoT and cloud eliminates the burden of local storage and supervision. However, cloud service providers can certainly gain control of users' data, which seriously threatens the security of data. As a result, integrity verification of IoT data is of great significance for effective cloud storage. Existing data integrity verification schemes for cloud storage are mainly based on hash functions [3], asymmetric cryptographic algorithms [4], and erasure codes [5]. Data integrity verification methods can also be divided into provable data possession (PDP) mechanism [4] and proofs of retrievability (POR) mechanism [6] according to whether it can correct wrong data after verification. These traditional methods often rely on trusted Third Party Auditors (TPAs) to execute auditing tasks and the burden of users during the verification phase can be decreased. For example, in Wise Information Technology of 120 (WIT120), massive electronic health records (EHR) are collected by wearable devices and then stored in the cloud. Before accessing health data, services providers usually offload the validation task to TPAs to guarantee data integrity. However, in real world scenarios, TPA is not completely trusted. Even with encryption methods [7] which can avoid the leakage of users' privacy, the quality and effectiveness is completely dependent on the credibility of TPA.

The associate editor coordinating the review of this manuscript and approving it for publication was Tie Qiu.

Blockchain technology has recently emerged as one of the most promising technologies and attracted great attentions for its transparency, immutability, security and decentralization. Researchers have considered to execute integrity verification services in the decentralized blockchain network, where transactions can be performed with no need of a trusted TPA. Ethereum and Hyperledger Fabric are two popular frameworks for the implementation of blockchain network [8]. However, there is a significant scalability barrier for blockchain related applications, which limits their capability to support services with large-scale and frequent transactions, e.g., the computational and communication overhead during integrity verification for large-scale IoT data [9]–[11]. Besides, dynamicity of IoT data [12], [13] has seldom been investigated for most of the existing blockchain based data integrity methods.

In order to address the problems above, we propose a Blockchain and Bilinear mapping based Data Integrity Scheme (BB-DIS) for large-scale IoT data in cloud storage. Main contributions of this paper are listed as follows:

- A blockchain based data integrity verification framework is proposed for large-scale IoT data. An associated series of protocols followed with verification algorithms and performance analysis are also presented in detail.
- A prototype system is built with an edge computing processor in the vicinity of the IoT devices to preprocess the large-scale IoT data so that communication cost and computation burden can be reduced significantly.
- Multiple simulation experiments are conducted on Hyperledger Fabric. Comparative analysis on computational and communication overhead among BB-BIS and other baseline schemes is given. Various sampling strategies are introduced, and optimized sampling verification scheme is finally recommended.

The rest of this paper is organized as follows: Section II gives an overview of related work in the literature. Section III elaborates on our proposed scheme from four parts: framework, protocol, algorithm and performance analysis of the algorithm. Section IV introduces the prototype system. Section V gives experimental results and analysis. Section VI makes the conclusions.

II. RELATED WORK

A. TRADITIONAL WORK ON DATA INTEGRITY

The concept of data integrity verification was first proposed by Deswarte *et al.* [3] in 2004. By calculating and comparing message authentication code (MAC) values, two solutions were proposed to determine whether the data on remote nodes was complete or not. However, the general communication overhead and computational cost were very large. On this basis, Sebe *et al.* [14] used the method of blocking the original data files to reduce the computational cost. Later, Ateniese *et al.* [4] first proposed the PDP scheme, which uses Rivest-Shamir-Adleman (RSA) signatures. The model generated probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduced I/O costs. The client maintained a constant amount of metadata to verify the proof. Its protocol was defined for static files and could not handle dynamic data storage without introducing security vulnerabilities. This problem was solved in [15], but did not support fully dynamic data operations.

Wang *et al.* [16] proposed a new challenge and response protocol, which used the Merkle hash tree to ensure the correctness of the data block and introduced an independent TPA instead of the user to execute the verification operation to alleviate burden. Juels and Kaliski, Jr. [6] first proposed a sentinel-based POR model, which added some ‘‘sentinel’’ data blocks to the stored data at random and used the erasure code to detect distorted data and downgrade them to storage with undefined quality of service. Shacham and Waters [17] used Boneh-Lynn-Shacham (BLS) signature mechanism to generate homomorphic verifiable tags (HVTs) based on Ateniese et al' s research, which reduced communication overhead while supporting public auditing. But it could not guarantee users' data privacy. Wang *et al.* [5] used the linear characteristics of erasure codes to achieve partial dynamic operations. Chen and Curtmola [18] used Cauchy Reed-Solomon linear coding to preprocess data to improve the recovery speed of erroneous data, but the computational cost was still very large. In order to prevent TPA from leaking privacy data [19], Wang *et al.* [7] proposed a data integrity verification mechanism based on public key-based homomorphic authenticator and random mask to achieve privacy protection in public cloud system. To reduce the energy consumption in wireless sensor networks (WSNs) [20], Ben Othman *et al.* [21] adopted a symmetric-key homomorphic encryption to protect data privacy and combines it with homomorphic signature to check the aggregation data integrity. Zhu *et al.* [22] reduced the computational overhead of the hash function in the signature process [23] and used the random masking technique to preserving data privacy.

In consideration of the particularity and complexity of graph database, Arshad *et al.* [24] presented two security notions based on hash message authentication code (HMAC) for graph data integrity verification and query results. Consider the direction of the edge, Reina *et al.* [25] calculated a new hash value e.g., chained hash from the concatenation of the current node.

B. APPLICATIONS OF BLOCKCHAIN

With its key characteristics of decentralization, persistency and auditability [26], blockchain technology has evolved as an enabling and disruptive technology that has been gradually adopted across many industry vertical domains. Blockchain can be a key enabler to solve many IoT security problems [2]. Suliman *et al.* [27] presented a blockchain-solution and implementation using Ethereum smart contracts for monetizing IoT data with automated payment involving no intermediary. Albreiki *et al.* [28] proposed a blockchain-based system using Ethereum smart contracts to manage access control policies for IoT data access in a decentralized manner without a trusted third party. However, the integrity problems of IoT data still need to be addressed. Chaer *et al.* [29] discussed the system integration architecture and sequence flow diagrams to illustrate how blockchain can support and facilitate 5G networks. Salah *et al.* [30] discussed how the integration of artificial intelligence (AI) and blockchain can help in developing a new ecosystem of decentralized economy and outlined open research challenges in leveraging blockchain features for future AI applications. In order to detect the fake digital contents, Hasan and Salah [31] provided a solution using Ethereum smarts contracts to trace and track the provenance and history of video content to its original source even if the content is copied multiple times.

C. BLOCKCHAIN BASED WORK ON DATA INTEGRITY

Based on previous research work on cloud storage service architecture and data integrity, Liu *et al.* [9] proposed a blockchain based approach for IoT data integrity service. This solution performed integrity verification without relying on any TPAs in a dynamic IoT environment. However, the speed of uploading IoT data and the size of the verified data need to be improved. Yue *et al.* [10] proposed a blockchain based P2P cloud storage data integrity verification framework. They used Merkle tree for data integrity verification, and analyzed system performance under different Merkle tree structures. Liang *et al.* [32] proposed a decentralized and trusted cloud data provenance to verify data security. The provenance auditor verifies provenance data through information in the block. Wang *et al.* [11] proposed a decentralized model to solve the single point of trust problem in the traditional data auditing service model by collective trust. The protocol allows users to trace the history of their data.

In summary, most of existing data integrity verification methods based on blockchain technology focus on trust problem instead of data size. A more notable question is that IoT data stored in the cloud need to be updated in real time to meet the latest requirement of various applications. Therefore, it is necessary to propose a blockchain based dynamic solution aiming at data renewal for data integrity verification.

III. OUR SCHEME BB-DIS

A. PRELIMINARY

1) BLOCKCHAIN TECHNOLOGY

Blockchain technology implements decentralized peer-to-peer transactions, coordination and collaboration without the need for trust, through data encryption, time stamping, and distributed consensus. It can address the problem of high cost, inefficiency, and insecure data storage of centralized systems. The first generation blockchain introduced by Bitcoin is the public ledger for digital currency transactions. Many researches about the blockchain are shielded by Bitcoin. But blockchain could be applied to a variety of fields far beyond Bitcoin [33]. The second generation blockchain provides a versatile programmable platform, and smart contracts are introduced as autonomous programs that are deployed and run in blockchain networks. Smart contract is a digital protocol that aims at establishing agreement between communicating parties based on predefined rules and without the need for a trusted third party [34]. Smart contracts can be used to represent triggers, constraint conditions, and even entire business processes. Ethereum, featuring smart contracts, is a popular second-generation blockchain. The transaction in Ethereum is a signed message initiated by an external account, transmitted by the Ethereum network, and recorded (excavated) on the Ethereum blockchain [35]. There are usually three types of transactions in Ethereum: transferring transactions, creating smart contracts, and execution of smart contracts. Hyperledger Fabric is a kind of permissioned blockchain, which provides a variety of consensus mechanisms. The 0.6 version and 1.0 version of Hyperledger Fabric provide PBFT and Kafka consensus mechanisms respectively. It has been widely used in the development of decentralized applications.

2) BILINEAR MAPPING

We assume that G_1 is a Gap Diffie-Hellman (GDH) group, P is the generator of group G_1, and G_2 is another multiplicative cyclic group whose order is prime q. The mapping $e : G_1 \times G_1 \rightarrow G_2$ is called bilinear pairing and has the following characteristics:

Computability: For arbitrary $a, b \in G_1$, there is an effective algorithm to calculate $e(a, b)$.

Bilinear: For arbitrary $x, y \in Z_q, a, b, c \in G_1$:

$$e\left(a^x, b^y\right) = e\left(a, b\right)^{(xy)},$$
$$e\left(a, bc\right) = e\left(ba, c\right) = e(a, b)e(a, c).$$

Non-degenerate: P is non-degenerate if $e\left(P, P\right) \neq 1$.

3) SHORT SIGNATURE

The signature bits of RSA, Digital Signature Algorithm (DSA) and BLS are 1024 bits, 320 bits, and 160 bits under the same security conditions. The computational security of RSA algorithm relies on the difficulty of factoring large integers and the RSA-based scheme has too much computational overhead. The DSA signature is a development of RSA signature, but it cannot be used to encrypt data files. The BLS signature can work in any bilinear cryptographic context and the protocol is unforgeable in the random oracle model. However, the BLS-based scheme needs to adopt a particular hash function which has efficiency issues for large-scale data. To this end, a secure hash function $H : \{0, 1\}^* \rightarrow Z_q^*$ is introduced in this paper. It can be a general cryptographic hash function such as SHA-1 or MD5.

ZSS short signature is based on a bilinear pairing proposed by Zhang *et al.* [23]. The signature system is less overhead than BLS signature. We assume that $e\left(P, P\right) \neq 1$, using the property of bilinear mapping, $e\left(P^x, P^y\right) = e\left(P, P\right)^{(xy)}$. It mainly contains three functions:

KeyGen. The data owner selects a random integer $\alpha \leftarrow Z_q^*$ as the private key sk, and αP as the public key pk. We cannot calculate α from pk.

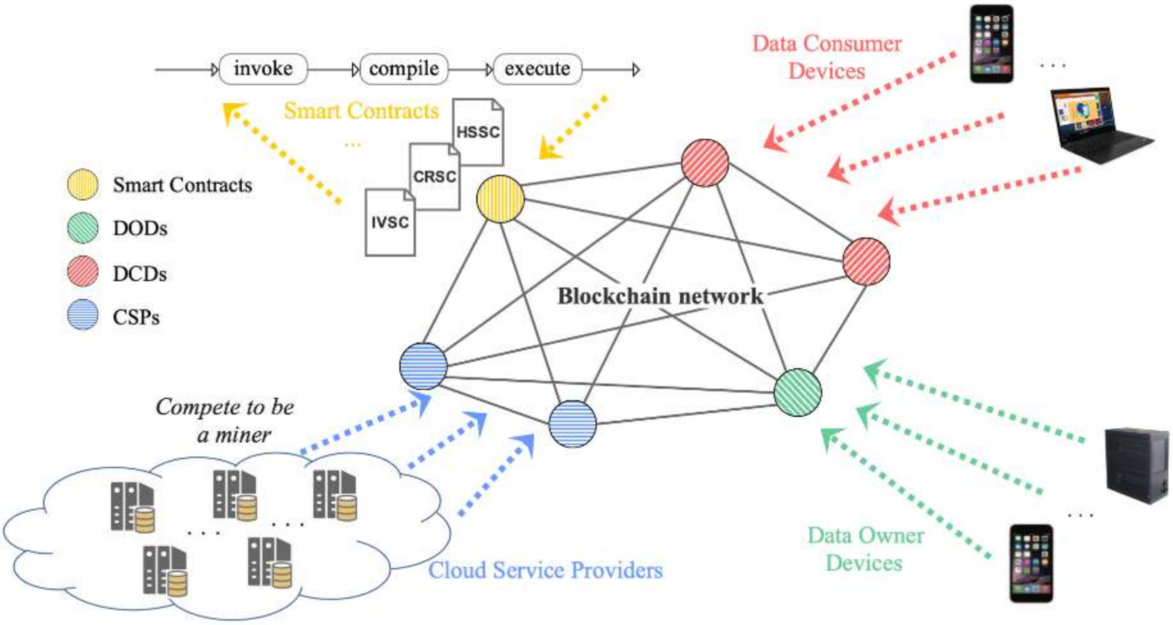


FIGURE 1. Framework of BB-DIS.

Sign. The signature of the message m is $Sig = \frac{1}{H(m)+\alpha}P$.

Verify. A verifier knows αP (pk), m, Sig', and needs to verify $Sig' = \frac{1}{H(m)+\alpha}P$, that is, calculate $e(P, P)$ and $e\left(H(m)P + \alpha P, Sig'\right)$ and judge whether they are equal. If they are equal, the signature is generated by the person who owns the private key α. The verification works because of the following equations:

$$\begin{aligned} e(H(m)P + \alpha P, Sig) &= e\left((H(m) + \alpha)P, \frac{1}{H(m)+\alpha}P\right) \\ &= e(P, P)^{(H(m)+\alpha)\cdot\frac{1}{H(m)+\alpha}} \\ &= e(P, P) \end{aligned} \tag{1}$$

B. FRAMEWORK OF OUR SCHEME

TABLE 1 gives a list of main symbols that will be covered in our scheme.

TABLE 1. Symbols in the framework.

Symbol	Definition
DOD	Data Owner Device
DCD	Data Consumer Device
CSP	Cloud Service Provider
HSSC	HVTs Storage Smart Contract
CRSC	Challenge Receiving Smart Contract
IVSC	Integrity Verification Smart Contract

The framework of BB-DIS is depicted in Figure 1, which mainly includes four kinds of entities, i.e., Smart Contracts, Data Owner Devices (DODs), Data Consumer Devices (DCDs), and Cloud Service Providers (CSPs). To achieve different functions, there are three kinds of smart contracts, i.e., HVTs Storage Smart Contract (HSSC), Challenge Receiving Smart Contract (CRSC) and Integrity Verification Smart Contract (IVSC). All of those entities can be acted as blockchain nodes in a blockchain network. In reality, data integrity verification involves multiple data owners and data consumers. The integrity verification is executed by smart contracts in a blockchain system. Users with integrity requirements can launch blockchain clients on their node devices or exit the blockchain network. The CSP also serves as a node in the blockchain network, which makes the nodes completely dispersed and the integrity verification more efficient.

DODs and DCDs should be added to the blockchain network when the blockchain system is initialized to generate a key pair. The data owner needs to pay for the interaction with the smart contract and the cloud storage service. CSP can act as a miner node in the blockchain network, which is qualified to provide services through mining and earn the corresponding reward. The data consumer requests to use the data stored in the cloud server and pays the corresponding expenses for it. Unlike Hyperledger Fabric, each node account must be set to have enough gas to make the transactions done successfully if we use Ethereum network. During each transaction, DOD pays promissory gas for data storage service to the corresponding CSP.

Each cloud service provider will provide cloud storage services, such as Amazon S3, IBM Bluemix, Microsoft Azure and Smart Ocean. In this framework, CSP provides a common data storage service for data owners, while non-cloud data can be transmitted over an inter-node P2P network.

Once deployed, smart contracts are difficult to modify, so if there are some security holes in smart contracts, it is difficult to prevent attacks by hackers. In this case, it is vital to test the smart contract code thoroughly and use essential security analysis tools to eliminate any security vulnerabilities [36]. *Chaincode Scanner* is a security analyzer for Hyperledger Fabric smart contracts. We can detect if there are security holes in the smart contracts with the analyzer.

For the sake of security and efficiency in the blockchain network, which are the two most concerns in this field, we make the following two assumptions. One is that 51% attack and selfish mining are rarely possible if all the participating nodes pursue benefiting themselves. The most obvious proof is Bitcoin blockchain, instead of initiating 51% attack, a malicious attacker would be more willing to use the computation power to mine. Actually, such attacks seldom occur. The other assumption is that blockchain consensus can be reached within a short time. A transaction can be validated with an average duration of 12 seconds in the Ethereum blockchain. Hyperledger Fabric is expected to enable consensus less than 1 second. Generally speaking, Hyperledger Fabric will be more suitable for enterprise-level blockchain applications compared with other blockchain platforms.

C. VERIFICATION PROTOCOL

The verification process of this scheme is shown in Figure 2 and the transactions between different smart contracts and all actors are listed in Table 2 in detail. The protocol is divided into three stages: step stage, challenge stage, and verification stage. Smart contract and CSP play the role of verifier and proof provider respectively.

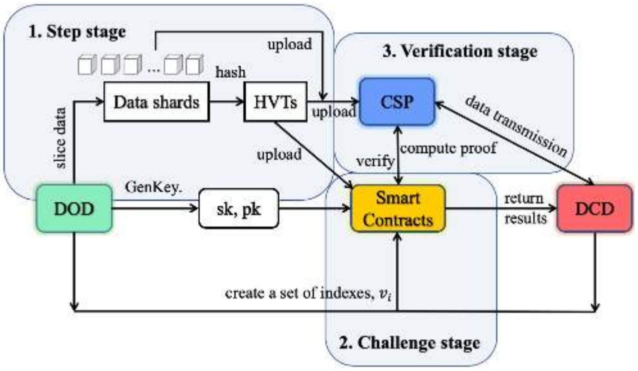


FIGURE 2. Verification protocol.

Step stage: The DOD establishes a bilinear mapping, selects a short signature hash function, selects a private key randomly and calculates a corresponding public key from the private key. Then, the DOD slices a data file into a set of several equal length data shards. After hashing, DOD calculates the HVT of each data shard to generate an authentication metadata set, which reduces communication overhead while supporting public auditing. The DOD uploads the data shards set and the metadata set to the cloud storage server. The metadata set is sent to the HSSC through the blockchain network in the form of a transaction. DOD deletes the data file locally.

Challenge stage: The DOD extracts c elements from HSSC to construct a data shard index set randomly, and sends a series of random values along with the data shard index set to the CSP and the CRSC in the form of *chal*, a challenge request.

TABLE 2. Protocol: Data integrity verification using blockchain.

Step	Entities	Operation
Step stage		
1	DOD	generate sk, pk (for short signature)
2	DOD	slice data into shards, generate HVTs
3	DOD → CSP	upload data shards and HVTs
4	DOD → HSSC	upload HVTs
Challenge Stage		
1	DOD	create a set of indexes $I = \{s_i\}$, v_i
2	DOD → CSP	$chal = \{(i, v_i)\}$
3	DOD → CRSC	$chal = \{(i, v_i)\}$
Verification Stage		
1	HSSC → IVSC	send HVTs
2	CRSC → IVSC	send *chal*
3	CSP	compute proof $\{R, \mu, \eta\}$
4	CSP → IVSC	send proof $\{R, \mu, \eta\}$
5	IVSC	verify
6	IVSC → DOD	return the verification result

Verification stage: IVSC gets HVTs and *chal* from HSSC and CRSC respectively. After receiving the challenge request, CSP computes the proof $\{R, \mu, \eta\}$ and sends it to the IVSC. The IVSC verifies whether the proof is correct. If it is correct, the data stored in the cloud is integrate and IVSC returns the result to DOD.

Actually, a DCD can also initiate a verification request for the stored data. Under this circumstance, a DOD is still needed for preliminary work in the step stage. A DCD requesting for integrity verification service can participate the challenge stage and verification stage. When the data integrity is confirmed, the CSP sends the data in servers to the corresponding client through the P2P network directly.

D. SC-VERIFICATION ALGORITHM

We use smart contracts in the verification process, and propose the verification algorithm according to the verification protocol to verify the authentication metadata. The step stage, challenge stage and verification stage of applying the SC-Verification algorithm are as follows:

Step stage: G_1 is a q-order cyclic addition group, P is one of its generators. G_2 is a q-order cyclic multiplicative group. Z_q stands for the integer ring of the mod q.

Firstly, DOD should establish a bilinear mapping:

$$e : G_1 \times G_1 \rightarrow G_2$$

and a short signature security hash function:

$$H : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$$

Given $\phi(i, j) : Z_q^* \times \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$ is a pseudo-random function where $k_0 \in Z_q^*$ and $|q| \geq \lambda \geq 160$.

DOD selects a private key $\alpha \leftarrow Z_q^*$ randomly, the corresponding public key is $Y = \alpha P$. The public key pk is Y and the private key sk is α. We can't calculate the private key from the public key.

DOD divides the data file F into data shards of equal length: $\{m_1, m_2, m_3, \ldots, m_n\}$, and generates a HVT for each data shard m_i:

$$\delta_i = \frac{1}{H\left(m_i\right) + \alpha} P \tag{2}$$

There is a collection of metadata: $\Phi = \{\delta_1, \delta_2, \ldots, \delta_n\}$.

Finally, DOD uploads the data shard set to cloud storage server and sends the metadata set Φ to HSSC. DOD deletes the data file locally. The process is illustrated in Algorithm 1:

```
Algorithm 1  Step
Input: {Data File $F$, Hash Function $H$}
Output: {$\delta_i$}
  $F = \{m_1, m_2, m_3, \ldots, m_n\}, \delta_i = 0$
    for $i = 1$ to $n$ do
      $\delta_i = {}^{P}/{H\left(m_i\right) + \alpha}$
    end for
  return $\delta_i$
```

Challenge Stage: The DOD extracts c elements randomly to construct a data shard index set $I = \{s_1, s_2, \ldots, s_c\}$, $c \in [1, n]$, and generates a pseudo-random number for each $i \in I$. DOD sends the random value and the data shard index set to CSP and CRSC in the form of the challenge request $chal = \{(i, v_i)\}, s_1 < i < s_c$.

Verification Stage: As a proof provider, after receiving the $chal$ request, CSP calculates:

$$R = \sum_{i=s_1}^{s_c} v_i Y \tag{3}$$

$$\mu = \sum_{i=s_1}^{s_c} v_i H(m_i) P \tag{4}$$

$$\eta = P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\delta_i} \tag{5}$$

CSP returns $\{R, \mu, \eta\}$ as the proof to the IVSC.

After receiving the proof $\{R, \mu, \eta\}$, IVSC calculates whether the data in the cloud storage server is integrated:

$$e\left(\eta, P\right) \cdot e\left(\mu + R, P\right) = e\left(P, P\right) \tag{6}$$

If the equation is true, the data is intact. The smart contract returns the verification result to the service requester. The process is illustrated in Algorithm 2:

```
Algorithm 2  Challenge and Verification
Input: {Data Block Index $I$, $\Phi$, Hash Function $H$, $P$, $Y$}
Output: {Proof $R$, $\mu$, $\eta$}
  $I = \{s_1, s_2, \ldots, s_c\}$
  $R = 0$
  $\mu = 0$
  $\eta = P$
    for $i = s_1$ to $s_c$ do
      $v_i = \phi(k_0, i)$
      $proof1 = v_i Y$
      $proof2 = v_i H(m_i) P$
      $proof3 = {}^{v_i}/_{\delta_i}$
      $R = R + proof1$
      $\mu = \mu + proof2$
      $\eta = \eta - P^2 \cdot proof3$
    end for
  return $R, \mu, \eta$
```

E. PERFORMANCE ANALYSIS OF SC-VERIFICATION ALGORITHM

1) FEASIBILITY

According to the scheme above, if the data stored by the CSP is unbroken, the proof sent by CSP is correct. The following calculation proves the correctness of our scheme.

$$\begin{aligned} & e\left(\eta, P\right) \cdot e\left(\mu + R, P\right) \\ & = e\left(P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\delta_i}, P\right) \cdot e\left(\sum_{i=s_1}^{s_c} v_i H(m_i) P + \sum_{i=s_1}^{s_c} v_i Y, P\right) \\ & = e\left(P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i(H\left(m_i\right) + \alpha)}{P}, P\right) \\ & \quad \cdot e\left(\sum_{i=s_1}^{s_c} v_i H(m_i + \alpha) P, P\right) \\ & = e\left(-\sum_{i=s_1}^{s_c} v_i H(m_i + \alpha) P, P\right) \cdot e\left(P, P\right) \\ & \quad \cdot e\left(\sum_{i=s_1}^{s_c} v_i H(m_i + \alpha) P, P\right) = e\left(P, P\right) \end{aligned} \tag{7}$$

From the deduction of equation (7), we can see that our sc-verification algorithm is feasible.

2) SECURITY

We assume that there are some attackers or malicious servers, tampering with the data stored by data owners in the cloud. If they want to pass the verification of the smart contract, then they need to construct the signature $\delta_j^* = \frac{1}{H\left(m_j^*\right) + \alpha} P$ to make:

$$\mu^* = \left(\sum_{i=s_1, i \neq j}^{s_c} v_i H(m_i) P\right) + v_j H(m_j^*) P \tag{8}$$

And we can get:

$$\eta^* = P - \left(P^2 \sum_{i=s_1, i \neq j}^{s_c} \frac{v_i}{\delta_i}\right) - P^2 \frac{v_j}{\delta_j^*} \tag{9}$$

$$e\left(\eta^*, P\right) \cdot e\left(\mu^* + R, P\right) = e\left(P, P\right) \tag{10}$$

However, neither the attacker nor the malicious server knows the private key α , so it is impossible to forge a m_j^* that satisfies:

$$\frac{1}{H\left(m_j^*\right)+\alpha}P=\frac{1}{H\left(m_j\right)+\alpha}P \tag{11}$$

and the proof cannot be modified.

If the attackers or malicious servers delete the data m_j in the cloud storage server, similar to the analysis above, it is impossible to forge a valid m_j^* because the private key α is unknown.

From the analysis above, we can conclude that our sc-verification algorithm can counter malicious attacks.

3) DYNAMICITY

The data dynamic update operation supported by our scheme is completed by update request algorithm *UpdateReq()* and update execution algorithm *UpdateExec()*. The corresponding operations include data shard appending, data shard modification and data shard deletion.

UpdateReq() : The algorithm runs on the DOD, requests an update execution of the outsourced file copy stored in remote CSP, and the output is an update request. The DOD sends the update request to the cloud in the form of $\langle BlockOp, Ind, m_i', \delta_i'\rangle$, where *BlockOp* is the corresponding data shard operation, and *Ind*, m_i', and δ_i' represent the index of the updated data shard, the updated data shard, and the updated metadata respectively.

UpdateExec(): The algorithm is executed on the CSP server. The input parameter is the update request of the DOD, and the output is a new file copy F' and a new metadata δ_i'. After each update, in order to ensure the correctness of the cloud update operation, DOD will execute the challenge agreement.

Appending operation: DOD inserts a new data shard in the position j. If there are n data shards initially, there will be $n+1$ data shards after the appending operation. If the generated challenge request contains the data block m_{n+1}, the verification can still be completed because the metadata set has been updated.

Deletion operation: When deleting a data shard, all subsequent data shards will be ahead of one position. If a specific data shard with an index value of j will be deleted, the DOD sends a delete request $\langle Delete, j, null, null\rangle$ to the CSP. After receiving the delete request, CSP deletes the data shard whose index position is j in the backups.

Modification operation: It's similar to the appending operation, and there will be n data shards after the insert operation.

IV. PROTOTYPE SYSTEM OF BB-DIS

Figure 3 shows a prototype system based on the BB-DIS. The system is divided into four layers from bottom to top, including 1) IoT devices, 2) edge computing devices and clients, 3) cloud storage service, and 4) data consumer devices.

As shown in Figure 3, each part of the system exists as a node in both the blockchain and the P2P network. IoT devices are responsible for data generation. Both clients and edge devices act as data owners. Edge computing is responsible for the processing and transmission of source data in the vicinity. Then, the processed IoT data is stored in the cloud server or smart contracts. The verification request will be initiated by the data owner client and a challenge request is sent through the blockchain network. The data consumer client can run on the PC and on the cloud, send data consuming requests or accept stored IoT data.

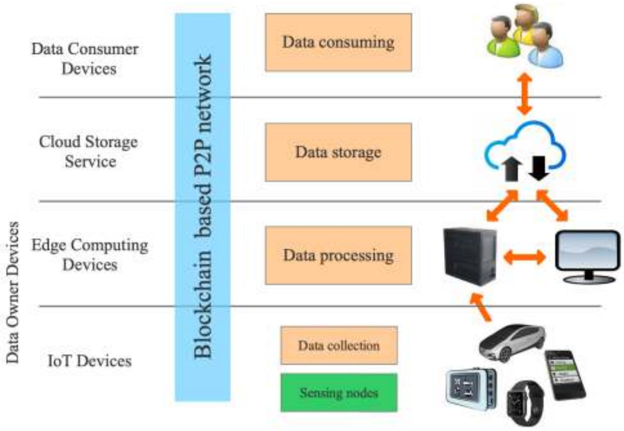


FIGURE 3. Prototype system.

A. EDGE COMPUTING

Edge computing is a decentralized architecture. Any node with computing resources and network resources between the data generation source and the cloud center is used as an edge node. Under this architecture, the operations of applications, data resources and services are moved from the central node to the logic edge node in the network. So, it can accelerate the processing and transmission speed of data, reduce delays, and make the processing of massive data more efficient [37]. With the rapid development of the IoT [38] and the promotion of cloud computing services, edge computing has brought us convenience in many aspects, such as: cloud offloading, video analytics, smart home, smart city, etc. [39], its features such as low latency and massive data processing greatly facilitate our lives.

Edge devices play an important role in the proposed blockchain based data integrity scheme. As shown in Figure 3, it can not only transport messages and transactions of IoT devices, but also help manage data storage and perform computations [40]. We list the functions of edge devices in our prototype system, as follows:

- Identify IoT devices. The edge server stores an identity copy of all IoT devices nearby and helps each device to generate data shards and HVTs.
- Create transactions for the IoT device. A valid blockchain transaction should include the signature of the IoT device or verification of signature from other nodes. So, we use edge servers to remedy IoT devices' faultiness.

- Collect and transfer data to the blockchain network. The edge server continuously collects data from nearby IoT devices. It finds the addresses which stand for cloud servers to store data and sends data blocks to them.

B. BLOCKCHAIN BASED P2P FILE SYSTEM

The P2P solution inherits the client-server model for small distributed environments where the server has powerful processing capabilities. Symmetric communication between peer nodes is the most obvious feature in P2P networks, and each peer node can become a client or server. The P2P system solves the bandwidth problem of sharing files from the server to the client. The peers can share files with each other through various parts without requesting all files from the server at the same time, which greatly enhances the scalability and efficiency of file sharing.

V. SIMULATION RESULTS AND PERFORMANCE EVALUATION

A. EXPERIMENTAL PREPARATIONS

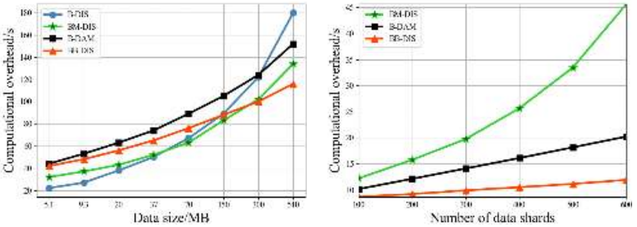
Basing on the prototype system, we set up a series of experiments to test the performance of our scheme. We use Inspur Yingxin NF8465M4 as the server, PC (Intel i7 quad-core processor 3.30GHz, memory 16GB) uses 64 bits operating system, the Blockchain platform is Hyperledger Fabric 1.1.0. Algorithms in this paper use a pairing-based cryptography (PBC) library version 0.5.14, and the key size is 160 bits, random number size is 80bits. The Raspberry Pi 3 B+ is used as the IoT device in the model to collect IoT data for integrity verification.

We set up an edge device based stream data processing structure near the data collection layer, and process the corresponding source data to generate the metadata set as the IoT data collection. We create a blockchain network on Hyperledger Fabric to provide a trusted environment for data integrity verification. The capacity of Ethereum is about 20∼30TPS, and the capacity of Hyperledger Fabric can exceed 4000TPS. In comparation with Ethereum and Bitcoin, in Hyperledger Fabric, smart contracts (Chaincodes) are always running on nodes but not stored in blocks and can implement a variety of complex business logic.

B. PERFORMANCE COMPARISONS

To prove the validity of our data integrity scheme (BB-DIS), we choose [9], [10] and [11] for comparative analysis, in which method [9] uses the blockchain network to store hash results directly. We call it B-DIS for short. Based on the blockchain [10], the data shards are hashed multiple times according to Merkle tree structure, we call it BM-DIS for short. And the method in [11] is called B-DAM for short because of its proposed Blockchain based data audit mechanism. The data in the experiment were taken as the average of 30 tests.

Figure 4 shows the consumption cost of integrity verification under different IoT data scales. We keep the total number of shards and the total number of samples constant. It can be seen from (a) that when data size is larger than 150MB, our scheme is more efficient. That is, it achieves trustless and greatly improves the speed of verification for large-scale data. As it is shown in (b), when the data shard size is fixed (20KB), BB-DIS takes less computational overhead than BM-DIS and B-DAM.

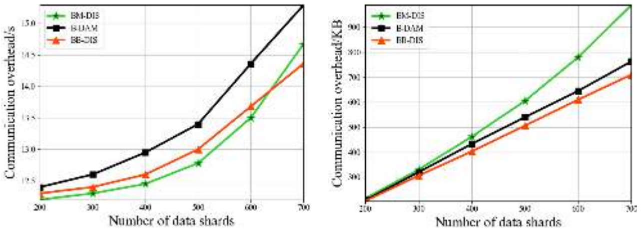


(a) Computational overhead with different data size.

(b) Computational overhead with different number of data shards.

FIGURE 4. Comparison of computational overhead.

The communication overhead refers to the time cost or the amount of data generated during data transmission between each part in verification process. The experimental results of three methods are shown in Figure 5. From (a) we can see that our solution's time cost become the smallest one when there are more than 700 data shards (50KB). It can be seen from (b) that as the number of sample data shards (1KB) increases, the space overhead increases linearly. Compared to the other two methods, our solution's space overhead is also the lowest. Hence, we can assert that our solution is more advantageous when the sample size is large.



(a) Time cost.

(b) Space overhead.

FIGURE 5. Comparison of communication overhead.

Figure 6 shows the time cost and accuracy with different number of samples for BB-DIS. The whole data size is fixed (10MB). And time cost grows as the number of checked shards increase. According to all the simulation results above, we can make a conclusion that our scheme has higher verification efficiency when the data size exceeds 300M. Meanwhile, our scheme has higher accuracy when the number of samples reaches 350, which is above 95%.

In order to prove the dynamics of our scheme, we carried out a series of related simulation experiments. The method in [10] also has dynamic property, but the author did not elaborate it in the article. We use it as a comparison object, and do simulation experiments on data appending and data

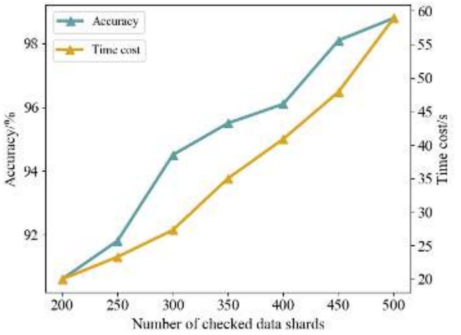
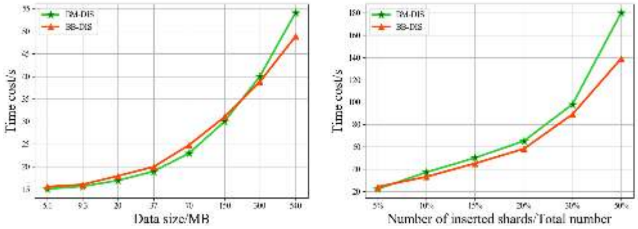


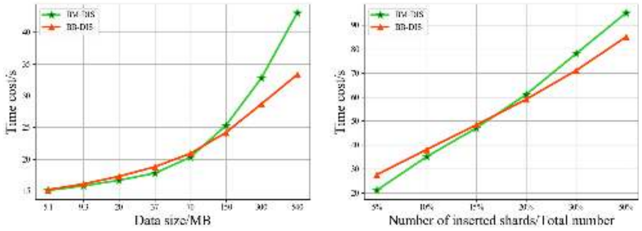
FIGURE 6. Time cost and accuracy of verification for BB-DIS.



(a) Time cost with different data size.

(b) Time cost with different number of inserted shards.

FIGURE 7. Comparison of time cost for appending.



(a) Time cost with different data size.

(b) Time cost with different number of inserted shards.

FIGURE 8. Comparison of time cost for modification.

modification operations, as shown in Figure 7 and Figure 8 respectively.

As illustrated in Figure 7 and 8, modification operation takes much less time than the appending operation, and the time it takes to modify the data depends on the time it takes for the short signature of the data shards. The time it takes for the appending operation depends on the speed of writing data to the disk. Obviously, as the data size increases, our solution performs well in terms of dynamic operation.

In addition, the algorithm does not calculate the time cost of deletion operation because it does not involve any computational overhead.

Next, we make a comparative analysis of BB-DIS and other three existing blockchain based data integrity methods B-DIS, BM-DIS and B-DAM in TABLE 3. We also discuss the main advantages of our proposed BB-DIS in the following.

Main advantages of BB-DIS are as follows:

TABLE 3. Performance comparison.

	Dyn.	Comm.	Comp.	
			CSP	IVSC
B-DIS	No	$O(t)$	$O(1)$	$O(1)$
BM-DIS	Yes	$O(c\log n)$	$O(c\log n)$	$O(c\log n)$
B-DAM	Yes	$O(n)$	$O(n)$	$O(n)$
BB-DIS	Yes	$O(n+c)$	$O(c)$	$O(c)$

Note: t denotes the number of data files. n denotes the number of data shards in each file. c denotes the number of shards being challenged (samples). ***Dyn.*** *denotes whether dynamic operations are supported.* ***Comm.*** *denotes communication cost.* ***Comp.*** *denotes computational complexity.*

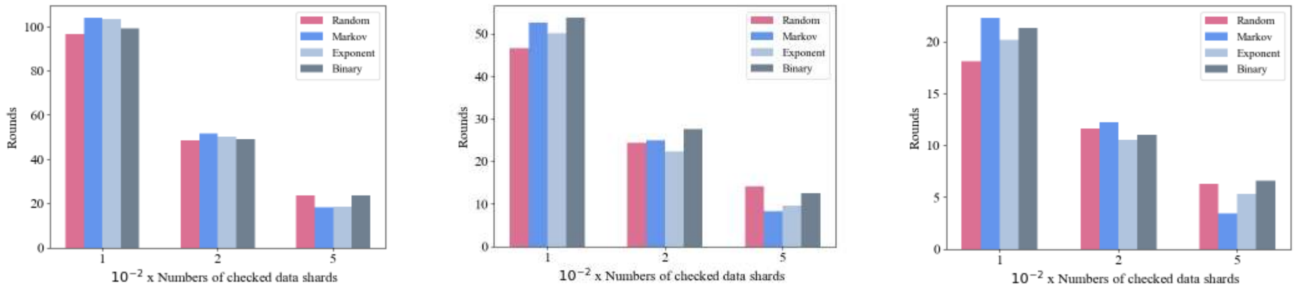
- It has a small communication overhead. Unlike BM-DIS, CSP does not need to carry auxiliary position information when transmitting data shards for verification in BB-DIS.
- It has a small verification delay. BM-DIS is affected by the structure of the Merkle tree when calculating the root node and needs to hash the data shards multiple times. The more layers a Merkle tree has, the greater cost we spend.
- It is no need for a particular hash function. Under such circumstance, the key size is only 160 bits. Although it needs extra preliminary work, it's outstanding when the data size is large.

C. SAMPLING ALGORITHMS

At present, most data integrity methods use simple random method for sampling and verification. The distribution function obeyed by the sample directly affects the sampling result. Therefore, we need to establish an optimal sampling model for the proposed data integrity scheme. Reference [10] deliberately invalidates a piece of sample data and compares the effects of several sampling models at different sample sizes. However, the corrupted data in the cloud server is likely to be far more than one. The original practical sampling model may not be applicable when the amount of corrupted data increases.

As shown in Figure 9, in our experiment, four sampling methods were compared: simple random distribution, Markov process sampling, exponential distribution sampling and binomial distribution sampling. The number of data shards is $n = 10000$, and corruption rate d is 0.01%, 0.02% and 0.05% respectively. The ordinate indicates the rounds when the data integrity is found to be damaged. In order to avoid contingencies, we executed 30 experiments and calculate the average value.

It can be seen from the experimental results that as the corruption rate increases, it takes less time to find the destroyed data. In our integrity verification model, when c is small, the effect of simple random distribution is better. When c reaches 500, the Markov process sampling has obvious superiority.



(a) Corruption rate d = 0.01%.

(b) Corruption rate d = 0.02%.

(c) Corruption rate d = 0.05%.

FIGURE 9. Comparison of different sampling algorithm.

VI. CONCLUSION

A data verification integrity scheme based on blockchain and bilinear mapping is proposed in this paper. Firstly, we combine smart contracts with bilinear mapping and propose a new data integrity verification framework. We slice the data into shards, and calculate metadata of each data shard for smart contract to execute verification. On this basis, the corresponding data integrity verification protocol and algorithm are proposed. We also introduce provable update mechanisms to deal with the dynamic property of IoT data in our scheme. Secondly, we propose a prototype system with an edge computing to process the IoT data. Experimental results finally demonstrate that the proposed BB-BIS outperforms existing blockchain based methods in terms of computational cost and communication overhead for large-scale IoT data.

Our future work will investigate on how to extend our scheme for more complex data types such as graph data and how to solve data recovery problems in large-scale IoT data.

REFERENCES

[1] F. Xiao, G. Ge, L. Sun, and R. Wang, "An energy-efficient data gathering method based on compressive sensing for pervasive sensor networks," *Pervasive Mobile Comput.*, vol. 41, pp. 343–353, Oct. 2017.

[2] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[3] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in *Proc. 6th Work. Conf. Integrity Internal Control Inf. Syst. (IICIS)*, 2004, pp. 1–11.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. CCS*, 2007, pp. 598–610.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in *Proc. IEEE 17th Int. Workshop Qual. Service (IWQoS)*, Jul. 2009, pp. 1–9.

[6] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in *Proc. CCS*, 2007, pp. 584–597.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[8] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2018.

[9] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proc. ICWS*, Jun. 2017, pp. 468–475.

[10] D. Yue, R. Li, Y. Zhang, W. Tian, and C. Peng, "Blockchain based data integrity verification in P2P cloud storage," in *Proc. ICPADS*, Dec. 2018, pp. 561–568.

[11] C. Wang, S. Chen, Z. Feng, Y. Jiang, and X. Xue, "Block chain-based data audit and access control mechanism in service collaboration," in *Proc. ICWS*, Jul. 2019, pp. 214–218.

[12] F. Xiao, X. Xie, Z. Jiang, L. Sun, and R. Wang, "Utility-aware data transmission scheme for delay tolerant networks," *Peer-Peer Netw. Appl.*, vol. 9, no. 5, pp. 936–944, Sep. 2016.

[13] H. Gao, Y. Duan, H. Miao, and Y. Yin, "An approach to data consistency checking for the dynamic replacement of service process," *IEEE Access*, vol. 5, pp. 11700–11711, 2017.

[14] F. Sebe, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, and J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[15] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. SecureComm*, 2008, Art. no. 9.

[16] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proc. ESORICS*, 2009, pp. 355–370.

[17] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. ASIACRYPT*, 2008, pp. 90–107.

[18] B. Chen and R. Curtmola, "Robust dynamic remote data checking for public clouds," in *Proc. CCS*, 2012, pp. 1043–1045.

[19] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 767–778, Apr. 2017.

[20] F. Xiao, L. Chen, C. Sha, L. Sun, R. Wang, A. X. Liu, and F. Ahmed, "Noise tolerant localization for sensor networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1701–1714, Aug. 2018.

[21] S. Ben Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and integrity for data aggregation in WSN using homomorphic encryption," *Wireless Pers. Commun.*, vol. 80, no. 2, pp. 867–889, Jan. 2015.

[22] H. Zhu, Y. Yuan, Y. Chen, Y. Zha, W. Xi, B. Jia, and Y. Xin, "A secure and efficient data integrity verification scheme for cloud-IoT based on short signature," *IEEE Access*, vol. 7, pp. 90036–90044, 2019.

[23] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2004, pp. 277–290.

[24] M. U. Arshad, A. Kundu, E. Bertino, A. Ghafoor, and C. Kundu, "Efficient and scalable integrity verification of data and query results for graph databases," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 866–879, May 2018.

[25] F. Reina, H. V. Netto, L. Rech, and A. F. Luiz, "A method to verify data integrity in graph databases," in *Proc. IEEE Symp. Comput. Commun.*, Jun. 2018, pp. 220–223.

[26] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jun. 2017, pp. 557–564.

[27] A. Suliman, Z. Husain, M. Abououf, M. Alblooshi, and K. Salah, "Monetization of IoT data using smart contracts," *IET Netw.*, vol. 8, no. 1, pp. 32–37, Jan. 2019.

[28] H. Albreiki, L. Alqassem, K. Salah, M. H. Rehman, and D. Svetinovic. (2019). *Decentralized Access Control for IoT Using Blockchain and Trusted Oracles*. [Online]. Available: https://www.researchgate.net/publication/335258940_Decentralized_Access_Control_for_IoT_Data_Using_Blockchain_and_Trusted_Oracles

[29] A. Chaer, K. Salah, C. Lima, P. P. Ray, and T. R. Sheltami. (2019). *Blockchain for 5G: Opportunities and Challenges*. [Online]. Available: https://www.researchgate.net/publication/335518169_Blockchain_for_5G_Opportunities_and_Challenges

[30] K. Salah, M. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10127–10149, 2019.

[31] H. Hasan and K. Salah, "Combating deepfake videos using blockchain and smart contracts," *IEEE Access*, vol. 7, no. 1, pp. 41596–41606, Dec. 2019.

[32] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, Madrid, Spain, May 2017, pp. 468–477.

[33] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[34] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct./Nov. 2018, pp. 1–8.

[35] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, Denver, CO, USA, Jun. 2016, pp. 181–194.

[36] H. Hasan and K. Salah, "Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*, vol. 6, pp. 65439–65448, 2018.

[37] J. J. Jung, "Computational collective intelligence with big data: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 66, pp. 87–88, Jan. 2017.

[38] M. M. U. Rathore, A. Paul, A. Ahmad, and G. Jeon, "IoT-based big data: From smart city towards next generation super city planning," *Int. J. Semantic Web Inf. Syst.*, vol. 13, no. 1, pp. 28–47, 2017.

[39] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[40] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, Mar./Apr. 2019.

HAIYAN WANG is currently a Professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, China. She is also the Deputy Director of the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing. She has hosted two National Natural Science Foundation of China projects. Her research interests mainly focus on services computing, trusted computing, big data intelligent processing technology, and blockchain technology.

JIAWEI ZHANG is currently pursuing the master's degree with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, School of Computer Science, Nanjing University of Posts and Telecommunications, China. His current research interests include services computing and blockchain technology.