Computer Standards & Interfaces 79 (2022) 103560

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

Toward secure distributed data storage with error locating in blockchain enabled edge computing

Dengzhi Liu[a,b,c], Yong Zhang[a,b,c], Dongbao Jia[a,c,*], Qiaosheng Zhang[a,c], Xuefeng Zhao[a,c], Huan Rong[d]

[a] School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China
[b] Jiangsu Institute of Marine Resources Development, Lianyungang 222005, China
[c] Jiangsu Engineering Research Center of Intelligent Port, Jiangsu Ocean University, Lianyungang 222005, China
[d] School of Artificial Intelligence, Nanjing University of Information Science & Technology, Nanjing 210044, China

A R T I C L E I N F O

Keywords:
Internet of things
Edge computing
Blockchain
Data storage
Error locating

A B S T R A C T

The technique of Internet of things (IoT) connects the distributed devices via the network that can realize smart applications, such as intelligent transportation, intelligent manufacturing, smart grid and smart home. Edge computing integrates the edge devices together and provides efficient computation and storage services with low latency for users. By combining with blockchain, edge computing can provide more secure data storage and transmission supporting tamper resistance and traceability for IoT. However, the existing data storage schemes are not suitable for processing the gathered data in blockchain enabled edge computing and the efficiency of the data error locating in the previous schemes is very low. In this paper, a secure distributed data storage scheme is proposed, which can be used in blockchain enabled edge computing. In the design of the proposed scheme, the technologies of bilinear pairing and BLS-HLA (BLS-Homomorphic Linear Authenticator) are used, which allows end clients to check the uploaded data storage. In addition, the mechanisms of error locating and data dynamics are supported due to the utilization of CBF (Counting Bloom Filter). Security analysis result indicates that the proposed scheme is correct, data blocks detectable and false positive rate negligible. Performance analysis result shows that the proposed scheme can be executed with low computational cost, which is practical for IOT environment in blockchain enabled edge computing.

1. Introduction

Internet of things (IoT) emerged as one important technique that can real-time collect the data information of surrounding environment and its own devices based on various equipped sensors [1]. The gathered data of IoT will be send to servers or other devices for further processing and storage via the Internet or LAN network. The concept of Internet of things is firstly put forward by ITU (International Telecommunication Union) in a report at the world summit on the information society (WSIS).[1] So far, many researches on the Internet of things have been proposed, such as data sharing, data aggregation and IoT nodes authentication [2–5]. Moreover, there are many IoT applications in real life. The typical IoT application scenarios include intelligent transportation, smart home, smart city, smart grid, etc.

The sensed data in IoT needs to be outsourced to remote powerful servers and then processed. The long distance of data transmission will cause response delay in data processing. The technique of edge computing can provide faster network service responses to IoT clients [7]. Edge computing is developed in the research of media, which can provide services of computation and storage for end users based on the nearby devices [6]. The feature of low latency of edge computing meets the requirement of real-time application in IoT [8]. However, edge computing has many disadvantages that cannot be directly used to process the IoT data, such as low computational capability, high dynamism and heterogeneity of the devices. Moreover, the edge devices is vulnerable to outside attacks due to the less of expertise on private-owned devices management. Hence, it is essential to incorporate other techniques to improve the service quality of edge computing.

* Corresponding author at: School of Computer Engineering, Jiangsu Ocean University, No. 59, Cangwu Road, Lianyungang 222005, China.
E-mail addresses: liudz@jou.edu.cn (D. Liu), zhyjou@jou.edu.cn (Y. Zhang), dbjia@jou.edu.cn (D. Jia), zqs@jou.edu.cn (Q. Zhang), zhaoxf@jou.edu.cn (X. Zhao), ronghuan@nuist.edu.cn (H. Rong).

[1] *The Internet of Things* Report: https://www.itu.int/internetofthings/

https://doi.org/10.1016/j.csi.2021.103560
Received 23 September 2020; Received in revised form 15 June 2021; Accepted 14 July 2021
Available online 20 July 2021

Blockchain is a technique of digital cryptocurrency that use community validation to synchronize the distributed ledgers [9–11]. By utilizing the technique of blockchain to design the storage and communication mechanism in edge computing, the system will be more secure in data processing and can improve the resource automation utilization [12–14]. Moreover, using the technique of blockchain in edge computing can improve the storage resources and computation capability of edge servers due to the property of consensus of blockchain. In recent studies of blockchain enabled edge computing, many valuable researches have been proposed [15–18], which greatly promotes the development of edge computing and IoT.

However, the issue of distributed data storage service was not considered in previous schemes. The security of IoT data storage in blockchain enabled edge computing determines whether the intelligence applications of IoT can be provided correctly [19]. In [20], Wang et al. proposed a dependable storage scheme to realize the secure distributed storage in cloud computing based on homomorphic encryption and erasure coding. To enhance the storage security, Wang et al. utilize the cryptography technology of random masking to propose a public storage auditing scheme combing the homomorphic encryption [21]. In [22], Wang et al. construct a dynamic storage scheme utilizing the technique of Merkle Hash Tree. Although some previous studies in other filed have focused on the research of storage security, the designed schemes cannot be directly used in blockchain enabled edge computing for distributed data storage. Moreover, there is no one scheme that can provide error locating and data dynamics simultaneously in edge computing. Hence, it is essential to design a secure distributed data storage scheme with error locating for IoT in blockchain enabled edge computing.

1.1. Contributions

A secure distributed data storage scheme with error locating is proposed for IoT in this paper. The proposed scheme can enhance the distributed storage security in blockchain enabled edge computing. The main contributions are listed as follows:

- The BLS-Homomorphic Linear Authenticator (BLS-HLA) is utilized to construct the proposed scheme, which enables the distributed storage can be checked by the end client without retrieving the data copy. Moreover, the data storage checking can be delegated to the edge servers, which highly alleviates the computational overhead of the local end side.
- Based on the technique of counting Bloom filter (CBF), a new data structure is designed, which names as CBT (Counting Bloom Filter Table). When the data storage checking fails, the edge server can execute binary search on the CBT to accurately locate error data block in the system.
- The proposed scheme can provide efficient data dynamics by updating the CBT. In this paper, the data dynamic operations that the proposed scheme supports including update, insertion, deletion and modification.

1.2. Organization

The rest of this paper is organized as follows: Section 2 provides some related works of this paper. Section 3 introduces the preliminaries of the proposed scheme. Section 4 presents the problem statement, including the system architecture and design objects of the proposed scheme. Section 5 introduces the detailed construction of the scheme in this paper. Section 6 evaluates the proposed scheme from the analysis of the security and the performance. Finally, this conclusion is given in Section 7.

2. Related works

In recent years, many researches of blockchain enabled edge computing have been proposed [15–18]. In [15], a blockchain enabled edge computing architecture is proposed by Liu et al., which can efficiently process the transportation data, enhance the data transmission security and save the energy in the system. In [23], Xiong et al. introduce a concept of mobile blockchain edge computing that can facilitate blockchain in mobile IoT system. To enhance the security of communication channel in edge computing, Zhang et al. utilize blockchain to design the authentication scheme for vehicles [16]. In Zhang et al.'s scheme, the route data is used to realize decentralized identification mechanism with privacy preservation by using a third party proxy. In [17], Xu et al. use blockchain to offload the computation in mobile edge computing. Moreover, the tool of nondominated sorting genetic is used in Xu et al.'s scheme that can generate strategies for balanced resource allocation. To process the large amount and heterogeneous streaming data, Liu et al. proposed a blockchain-based edge computing architecture to construct the video trading market in mobile edge computing [18]. Note that the entities in the system can adjust their execution ways according to the transactional data in Liu et al.'s scheme.

In the research of data storage security in edge computing, Xu et al. proposed a method by using the technologies of fountain code and XOR encryption [24]. In Xu et al.'s scheme, data file in the system is encrypted using XOR and then divided into multiple blocks. Then, the data blocks are mixed and distributed on different edge devices, which can efficiently protect the security of original data file. In [25], Zhang et al.'s proposed a VANET edge computing architecture based on blockchain to ensure the security of the data in the system. Three layers are included in the architecture, and the blockchain is used to improve the storage security of the data with the assist of the cloud server. In [26], Fu et al. proposed a secure scheme for industrial Internet of things to store the distributed data. In Fu et al.'s scheme, the gathered data in the system is processed by the edge server. In particular, the time-sensitive data is stored locally and other data is sent to the cloud server for long-term storage. Despite many secure storage mechanisms have been proposed in edge computing, the client cannot check whether the data storage in the system. In addition, the distributed data storage schemes should provide the error locating method for users, which highly improves the accuracy of data services provided by edge computing in IoT.

3. Preliminaries

The preliminaries of this paper are presented in this section, which includes bilinear pairing, homomorphic linear authenticator (HLA) and Bloom filter. The detailed description is shown as follows.

3.1. Bilinear pairing

Assume that there are two multiplicative groups $\mathbb{G}$, $\mathbb{G}_T$. The order of the two groups is q. Moreover, a bilinear map is computed as e: $\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ in the system. Select two points $\mathscr{P}$, $\mathscr{Q}$ from group $\mathbb{G}$. At the same time, randomly choose two integers of x, y from $\mathbb{Z}_q^*$. The symbol of $\mathscr{G}$ is the generator of $\mathbb{G}$. For any $\mathscr{P}_1$, $\mathscr{P}_2\in\mathbb{G}$, we have $e(\mathscr{P}_1\cdot\mathscr{P}_2, \mathscr{Q})= e(\mathscr{P}_1, \mathscr{Q})\cdot e(\mathscr{P}_2, \mathscr{Q})$. The bilinear map has the following three properties:

- Bilinear: $e(\mathscr{P}^x, \mathscr{Q}^y)= e(\mathscr{P}, \mathscr{Q})^{xy}$.
- Non-degenerate: $e(\mathscr{G}, \mathscr{G})\neq 1$.
- Computable: $e(\mathscr{P}, \mathscr{Q})$ can be computed by an algorithm.

3.2. Homomorphic linear authenticator

Homomorphic linear authenticator is a public key cryptography technique [27,28], which allows the end client delegates the storage checking tasks to a third party without revealing the private key and retrieving the original copy of the data. In the previous storage checking schemes [21,22], the BLS signature [29] is used to generate HLAs that computes a shorter signature for the data block. For a data file $F = (m_i, \ldots, m_n)$, the data owner can compute the authenticator for every data block as $A_i = (h(m_i))^a$, where a is the secret key and h is a one-way hash function. The data storage can be checked based on the bilinear pairing using public key $pk = \mathscr{G}^a$. The storage checking equation is $e(\prod h(m_i), pk) = e(A_i, \mathscr{G})$. If the checking equation holds, it means that the checked data blocks are stored correctly in the server.

3.3. Bloom filter

The technique of Bloom filter is proposed by Bloom [30] to check whether an element exists in a set. In the Bloom filter, k hash functions can map an element to the array bucket in the set. In the setup phase, the bucket values in the filter are set to 0. When the system adds an element, the positions that the element mapped using hash functions will be set to 1. The structure of a Bloom filter is shown in Fig. 1. If the system inputs an element ω and checks whether ω exists in the system, the system firstly needs to get the array positions of ω using k hash functions. If all the buckets of the positions are 1, it can be determined that element ω exists in the set with a high probability. In the network environment, the bloom filter is often used to filter spam, speed up the query process and spell checking. However, an inserted element in Bloom filter cannot be deleted, and the false positive rate of a checked element in the Bloom filer will be increased if the amount of the elements is large in the system.

Counting boom filter (CBF) To support the operation of data deletion, Fan et al. proposed a new Bloom filter structure that add a counter in each array bucket [31]. The CBF enables filter to support deletion operation by increasing the storage space. When a new joined element is inserted in the filter, the corresponding counter value in the mapped position will add 1. When the system needs to delete an element, the k counter values of the element in the *CBF* will subtract 1, respectively.

4. Problem statement

The problem statement of this is given in this section, including the system architecture and the design goals. The detailed introduction the problem statement is shown in the following part.

4.1. The system architecture

In the proposed scheme, the system architecture is mainly composed of three layers: IoT networks, blockchain enabled edge computing and cloud computing. Note that the data generated by the IoT networks in the system are uploaded to edge computing. By utilizing the technique of the blockchain, the data processing, transmission and storage will be more secure and reliable. Then, the distributed and fragmented data in edge computing can be gathered and stored in cloud computing, which can realize more powerful and long term data processing and storage. The system architecture is shown in Fig. 2. The detailed description of every layer and entities in the system is shown as follows.

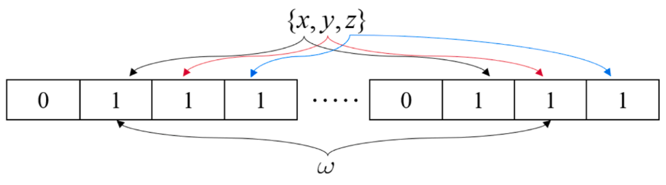

Fig. 1. The structure of the bloom filter.

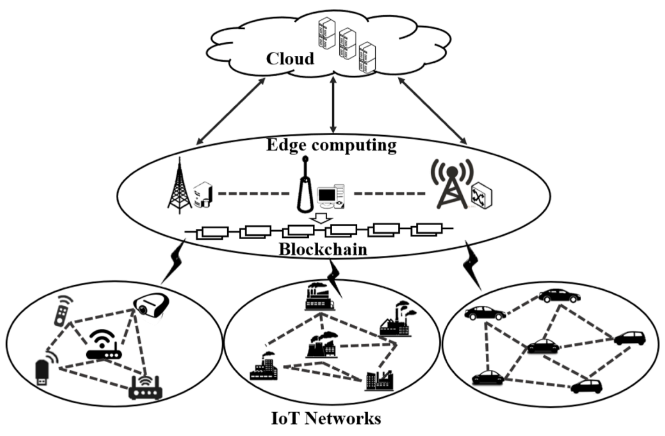

Fig. 2. The system architecture.

1. **IoT Networks:** In the IoT networks, many end devices, sensors and actuators are connected by the wireless and wired networks. Due to the limited resources of the storage and the computation, the end devices need to forward the continuously collected data to the up layer edge server or the other nearby devices. The received data from the other devices will be sent to the server or further forwarded by another devices. As the end client of IoT, the IoT devices and nodes are trusted in the data collection and transmission. Meanwhile, the end client of IoT is vulnerable to malicious attacks for data interception, tampering and other purposes driven by financial interests.
2. **Blockchain Enabled Edge Computing:** Edge computing can provide data processing, storage and computation for the end devices in real-time with low latency. The core idea of edge computing is to use the resources of equipments and computers around the end client to provide a higher quality services. However, the heterogeneity of the edge servers determines that edge computing suffers from jamming attacks and sniffer attacks. Due to the dynamicity of the edge servers, the integrity and reliability of data cannot be guaranteed. The technique of blockchain can realize distributed data storage, P2P transmission and consensus mechanism. By integrating the blockchain into edge computing, the data security in edge computing can be well enhanced.
3. **Cloud Computing:** Cloud computing in the system can provide long term data storage, data backup and powerful data processing for edge servers and end clients. Compared to the edge servers, the cloud servers are more centralized [32]. Moreover, due to the centralized management of the distributed servers, it is easier to maintain the rights and interests of users when problems arise in data storage and processing. In addition, cloud computing can provide unrestricted resource of storage and computation to users, which can make up for shortcomings of edge computing.

4.2. Design objects

In order to realize the secure distributed data storage in blockchain enabled edge computing, the proposed scheme in the design should achieve the following objects:

1. **Storage Checking Delegation:** To reduce the local communication and computational overhead, the end client can delegate the data storage checking tasks to edge servers.
2. **Storage Correctness and Completeness:** The distributed data storage correctness and completeness should be guaranteed that allows end client to check the storage regularly.
3. **Error Data Locating:** The proposed scheme should provide an efficient method for error data locating if the storage checking fails.

4. **Data Dynamics:** The data dynamics should allow the end clients to execute efficient dynamic operations on their stored data, including update, insertion, deletion and modification.

5. The proposed scheme

In the introduction of the proposed scheme, a high-level description is provided, which describes the main process of the proposed scheme. Then, the construction is described in detail.

5.1. High-level description

The proposed scheme mainly includes five phases, including *KeyGen*, *CBTGen*, *AuthGen*, *Check* and *Dynamics*. In the phase of *KeyGen*, the secret and public keys are generated, which will be used to compute the data file checking metadata. The phase of *CBTGen* is responsible for generating the structure of CBT according to the data. In the phase of *AutheGen*, the data file tag and the data block authenticators are computed, which will be sent to edge computing for the further storage. In the phase of *Check*, if the end client wants to check the distributed storage, it needs to compute the challenge information for the storage checking and send the challenge to the edge servers. Then, the edge server requests the cloud to compute the corresponding storage proof and check the proof using the public key. When the storage checking fails, the edge server can efficiently locate the error data according to the CBT. In the phase of *Dynamics*, the end client can execute the dynamic operations on the stored distributed data based on the CBT update.

5.2. The detailed construction

The detailed introduction of the phases in the proposed scheme is provided. The primary notations and the corresponding description are listed in Table 1. The detailed introduction of the proposed scheme is shown as follows:

1. $KeyGen(1^{\theta})\rightarrow(sk,pk)$: In the proposed scheme, the system randomly selects two multiplicative cyclic groups of $\mathbb{G}$, $\mathbb{G}_T$ with large prime order q. The generator of group $\mathbb{G}$ is $\mathscr{G}$, and the bilinear pairing e can be denoted as $\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$. he data file of one end client gathered by the sensors is $F=(m_i,\dots,m_n)$ in the system. Let H_0 be a hash function that inputs a value and outputs a point on the elliptic curve. Moreover, the system selects k hash functions H_1, …, H_k that are used to construct the CBT. In particular, the end client chooses a value of x from $\mathbb{Z}_q$ as the secret key. The corresponding public key is computed as $pk = \mathscr{G}^x$. Then, the end client selects a security parameter P from group $\mathbb{G}$ for the further data block authenticator computation.
2. $CBTGen(\{m_i\})\rightarrow CBT$: In the proposed scheme, the counting bloom filter (CBF) is used to index the data generated by the IOT networks. In the system, the end client can generate a counting bloom filter table (CBT) according to its data blocks before the data file uploading. Note that the operation of data block deletion and insertion can be realized by adding the counter in the bloom filter structure. For the end client, one specific data block can be quickly located by structuring its data blocks in a CBT. The CBT of the data is stored in edge computing. The edge server manages the CBT structure according to the stored data blocks. The CBT of the data file is illustrated as Fig. 3. Note that the red number of 2 and 3 is the counter of bucket 1 and bucket *p*-2, respectively. The algorithm of the data block insertion in the CBT is shown as Algorithm 1.
3. $AuthGen(\{m_i\},sk,pk)\rightarrow\Phi$: For the data block m_i, the authenticator can be computed as $A_i=(H_0(Fname)\cdot P^{m_i})^x$ using the secret key, where $\Phi=\{A_i\}_{1\le i\le n}$ and $Fname$ is the unique identifier of the data file F. Moreover, the end client generates a signature secret/public key pair as $\{ssk,psk\}$ and computes the data file tag as $t = Fname \parallel sig_{ssk}(Fname)$. Here, $sig_{ssk}(Fname)$ is the signature of $Fname$ under the secret key of the signature. Finally, the end client upload $\{F,\Phi,t\}$ and CBT to edge computing for the further processing based on the technique of blockchain. Due to the restricted resource of the storage, the distributed edge servers gather all the data files and the corresponding checking metadata, and outsource them to cloud servers.
4. $Check(\{m_i\},A_i,t,pk)\rightarrow 1/\perp$: In the proposed scheme, edge computing is constructed combined the technique of blockchain. The blockchain enabled edge computing can process and store the client's data file more securely. To allow the client to remotely check the data storage, in this paper, the storage checking tasks of the end client are delegated to the edge servers. When the end client needs to get the storage status of its c $(1\le c\le n)$ data blocks, it will selects c elements of $\{s_i\}_{1\le i\le c}$ and chooses v_i from $\mathbb{Z}_q$ for every s_i. Then, the end client sends $\{i,v_i\}$ to edge servers for the storage checking. Upon receiving the checking request, the edge server retrieves the corresponding data tag and verify its validity using the public key of the signature. If the verification outputs 1, the edge server recovers *Fname* from the tag; otherwise, the system outputs $\perp$. After that, the edge server sends $\{i,v_i\}$ to the cloud for storage checking. Upon receiving the checking request, it computes $\mu=\sum_{i\in s_i}m_iv_i$ and $A=\sum_{i\in s_i}A_i^{v_i}$. The proof of the storage can be denoted as $Proof_c=(\mu,A)$. Finally, $Proof_c$ will be sent to the edge server for the further storage checking. In the storage checking, the edge server can determine the storage correctness on behalf of the end client through checking Eq. (1).

$$e(A,\mathscr{G})=e\left(\left(\prod_{i\in c}H_0(Fname)^{v_i}\right)\cdot P^{\mu},pk\right) \tag{1}$$

When the storage checking outputs 1, it determines that the checked data blocks are stored correctly; otherwise, one or more data blocks are stored incompletely. The edge server can execute binary search under Algorithm 2 to locate the error data.
5. $Dynamics(\{m_i\},A_i,t)\rightarrow(\{m_i^*\},A_i^*,t^*)$: The data dynamics in the proposed scheme includes update, insertion, deletion and modification.

Table 1
Notations.

Symbol	Description
F, m	The data file and data block
H_0, …, H_k	The one-way hash functions
p	The total number of array buckets in CBT
n, k	The number of data blocks and hash functions in CBT
$\mathbb{G}$, $\mathbb{G}_T$	Two multiplicative groups
$\mathscr{G}$	The generator of group $\mathbb{G}$
$\mathbb{Z}_q$	The integer set with 0 to $q-1$
q	The large prime order
x, pk	The secret/public key of the end client
A_i, Φ	The authenticator of m_i and the authenticator set
t	The data file tag
ssk, psk	The secret/public key of the file identifier's signature
c	The challenging data block number
s_i, v_i	The elements for the storage proof generation
μ, A	The data storage proof

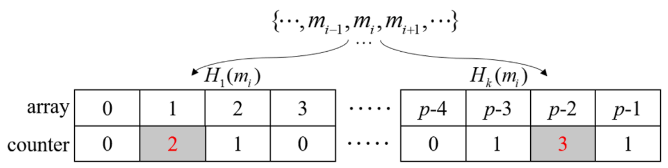

Fig. 3. The proposed CBT structure.

```
if $1 \le i \le n, 1 \le j \le p$ then
    $Bucket[j] = H_j(m_i)$
    $Counter[j] = Counter[j] + 1$
    return done
end if
```

Algorithm 1. Insertion.

```
if $1 \le i \le c, 1 \le j \le p$ then
    Define $l$=1, $r$=c
    if $l \le r$ then
        $Mid = (l + r)/2$
        if $Counter[j] \ne 0$ then
            if $Bucket[H_j(m_i)] \ne H_j(m_i)$ then
                return $Mid$
            else
                $Mid = (Mid + r)/2$
                return
            else
                $Mid = (l + Mid)/2$
                return
            end if
        end if
    end if
end if
```

Algorithm 2. Lookup.

The main process of the data dynamics in this paper is described as follows.

- *Update*: In the block update, the end client sends updated block of m_i^* with the corresponding checking metadata $\{A_i^*, t^*\}$ to the edge server. Moreover, when the edge server receives the updated data block, it will compute $H_j(m_i)$ and updates the CBT.
- *Insertion*: If the client wants to insert the index of m_i^* at the ith bucket in the CBT. Firstly, the edge server locates the block insertion position of $Bucket[i]$ and then executes Algorithm 1 to insert the corresponding data block. The corresponding checking metadata $\{A_i^*, t^*\}$ of inserted block m_i^* is also sent to the edge server for the further data storage checking.
- *Deletion*: If the data block m_i^* needs to be deleted, the edge server should locate the index passion of m_i^* in the CBT. Then, the corresponding counter of the ith bucket needs to be changed. The process of block deletion in the CBT can be found in Algorithm 3.
- *Modification*: In the data block modification, if the end client wants to modify one data block, it should request the edge server to delete the corresponding data block and then insert the modified data block information. That is to say, the edge server firstly executes Algorithm 3 to delete the index information of the deleted data block in the CBT. Then, the end client request the edge server to execute Algorithm 1 to inset the modified data block and its checking metadata.

```
if $1 \le i \le n, 1 \le j \le p$ then
    $Bucket[j] = H_j(m_i)$
    $Counter[j] = Counter[j] - 1$
    return done
end if
```

Algorithm 3. Deletion.

6. Evaluation

The security and the performance are analyzed in this section. In the security analysis, the correctness, the detectability and the effectiveness of the proposed scheme are proved. In the performance analysis, the efficiency of the proposed scheme is provided.

6.1. Security analysis

Theorem 1. *The proposed storage checking and the location mechanism can be proved to be correct.*

Proof of Theorem 1.. In the correctness proof, we need to assume that the end clients, edge servers and the cloud can compute the required parameters according to the proposed scheme. Moreover, all the parameters are complete in the transmission. In the storage checking, the edge server can check whether the data is stored correctly according to Eq. (1). If the above assumptions are true, the left side of the equation can be elaborated as follows:

$$\begin{aligned} & e(A, \mathscr{G}) \\ & = e\left(\sum_{i \in s_i} A_i^{v_i}, \mathscr{G}\right) \\ & = e\left(\sum_{i \in s_i} ((H_0(Fname) \cdot P^{m_i})^x)^{v_i}, \mathscr{G}\right) \end{aligned}$$

The right side of Eq. (1) can be solved using the end client's public key and the generated security parameter as follows:

$$\begin{aligned} & e\left(\left(\prod_{i \in c} H_0(Fname)^{v_i}\right) \cdot P^{\mu}, pk\right) \\ & = e\left(\left(\prod_{i \in c} H_0(Fname)^{v_i}\right) \cdot P^{\sum_{i \in s_i} m_i v_i}, \mathscr{G}^x\right) \\ & = e\left(\prod_{i \in c} (H_0(Fname)^{v_i} \cdot P^{m_i v_i}), \mathscr{G}^x\right) \\ & = e\left(\prod_{i \in c} ((H_0(Fname) \cdot P^{m_i})^{v_i}\right)^x, \mathscr{G}) \end{aligned}$$

That is to say, Eq. (1) can be proved to be hold. Therefore, the proposed storage checking is correct in this paper.

In the error data block locating mechanism, when the storage checking fails, the edge server can determine the error data location according to check the CBT. If the counter of the checked array is not 0, it can be determined that at least one index hash value can map to this bucket. Through checking whether the bucket value equals to the hash value of the corresponding data block, we can determine that the block stores in the cloud server. Hence, the proposed data block locating mechanism is correct. □

Theorem 2. *The corrupted or deleted data blocks in the proposed scheme is detectable.*

Proof of Theorem 2.. Form the description in the proposed scheme, we can get that the data block number of the data file is n. In the storage checking, the end client request the edge server to check c data blocks. Assume that w data blocks were deleted or corrupted by adversaries in the cloud, where $1 \le w \le c$. According to the c challenging data blocks, the probability of detecting at least one deleted or corrupted data block in the cloud is P_1. Assume that the probability that the deleted or corrupted data blocks cannot be detected is P_0, we can get that $P_1 = 1 - P_0$. The computation of P_0 is shown as follows.

$$P_0 = \frac{n - w}{n} \times \frac{n - 1 - w}{n - 1} \times \ldots\ldots \times \frac{n - c + 1 - w}{n - c + 1}$$

From the computation result of P_0, we can get that

$$\left(\frac{n-c+1-w}{n-c+1}\right)^c \leq \frac{n-w}{n} \times \frac{n-1-w}{n-1} \times \ldots\ldots \times \frac{n-c+1-w}{n-c+1} \leq \left(\frac{n-w}{n}\right)^c$$

Hence, the minimum probability of detecting the deleted or corrupted data blocks under c blocks challenging is $1-\left(\frac{n-w}{n}\right)^c$, and the corresponding maximum probability is $1-\left(\frac{n-c+1-w}{n-c+1}\right)^c$. □

Theorem 3. *The false positive rate of one inserted element in the CBT is negligible.*

Proof of Theorem 3.. Suppose that the probability of setting each counter is the same. The number of the index hash function is k. For a bucket in the array, the probability that the system does not calculate the corresponding counter value as required is $\left(1-\frac{1}{p}\right)^k$. Assume that all the data blocks are indexed in the CBT, the probability that the value of the counter in one bucket remains unchanged is $\left(1-\frac{1}{p}\right)^{kn}$. Hence, the false positive rate of one inserted element can be computed as $\left(1-\left(1-\frac{1}{p}\right)^{kn}\right)^k$. In practice, the number of data blocks and index hash function is very large. That is to say, the probability value of $\left(1-\left(1-\frac{1}{p}\right)^{kn}\right)^k$ is negligible. □

6.2. Performance analysis

To show the computational cost of the proposed storage checking, we simulate the main phases of the proposed scheme. The cryptographic operations are simulated on a PBC (Paring Based Cryptography) [34] and GMP (GNU Multiple Precision Arithmetic) [33] constructed experimental platform. The simulation computer is configured with 8 GB RAM and Intel 2.60 GHz CPU. The CBT is constructed on a laptop installed with MATLAB R2018b. The laptop is configured with Intel Core 1.8 GHz CPU and 8G RAM.

Fig. 4 demonstrates the simulation result of the proposed storage checking, including *KeyGen*, *CBTGen*, *AuthGen* and *Check*. As is shown in Fig. 4, the time cost of *KeyGen* remains approximately constant under the different block number. The time cost of the other three processes will be higher with the growth of blocks. Moreover, the time increase rate of *AuthGen* is the largest compared to that of *CBTGen* and *Check*. The main reason is that *AuthGen* needs to compute the tag and the authenticators for the data file. In addition, when data block number is the same, the time cost of *Check* is almost 1.5 times of that of *CBTGen*. The additional time cost in *Check* mainly results from the bilinear pairing in the storage checking. When the data block number is 100, the largest time cost of *AuthGen* is about 1700 ms, which is acceptable in practical application. Hence, the proposed storage checking is efficient and can be well used in practical blockchain enabled edge computing for IoT data storage.

To demonstrate the efficiency of the error locating mechanism, we simulate the time cost of the error locating under different accuracy ratio. Fig. 5 illustrates the simulation result. The comparative method is the traditional binary search in data locating in researches [35,36]. The time cost of the traditional binary search and the proposed method will be increased when the accuracy ration is higher. Note that the proposed method costs less time compared to traditional binary search under the same accuracy ratio. Moreover, when the accuracy ration requirement is about 100%, the time cost of the proposed method reduced by nearly 40% compared to that of the traditional binary search. Hence, the proposed error locating method in this paper is more efficient compared to previous methods.

Fig. 6 is the simulation result of the proposed dynamics in this paper. According to the simulation result, we can see that the four phases' time cost increases linearly with data block number. The growth rate of *Update* is the smallest, and the data growth rate of *Modification* is the largest. From the above description, we can conclude the reason is that *Modification* needs to delete the operated data block firstly and then execute the insertion operation to realize block modification. In addition, the process of deletion needs much time cost compared to *Insertion* when processed data blocks are same. Note that the largest time of the four phases is about 7 s, 5.8 s, 3 s and 1.2 s, which are still within the acceptable level and can be used to operate the data blocks for IoT in blockchain enabled edge computing.

7. Conclusion

With the development of edge computing and blockchain, the research on blockchain enabled edge computing has attracted the attention of researchers at home and abroad. To enhance the data storage in edge computing, in this paper, a secure distributed data storage is proposed for IoT in blockchain enabled edge computing. Due to the utilization of the techniques of bilinear pairing and BLS-HLA, the proposed scheme supports uploaded data storage checking. With the assist of the counting bloom filter (CBF), a new data structure of CBT is designed. The CBF can be used to realize data dynamics and error data locating when the storage checking is failed. In order to alleviate the end client side's computational overhead, the data storage checking tasks are delegated to the edge servers. Because the blockchain is used to construct the edge computing, the data processing and storage is more

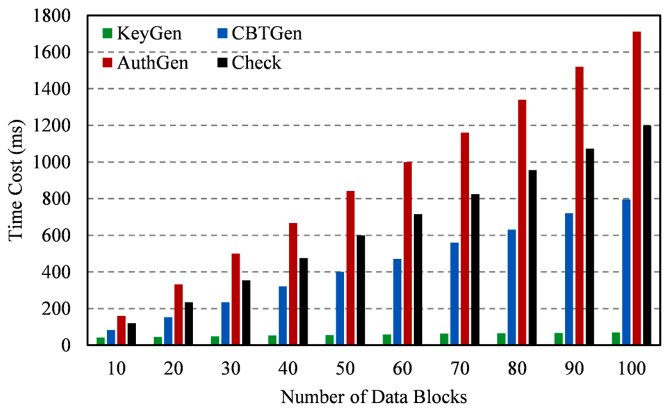

Fig. 4. The computational cost of the proposed storage checking.

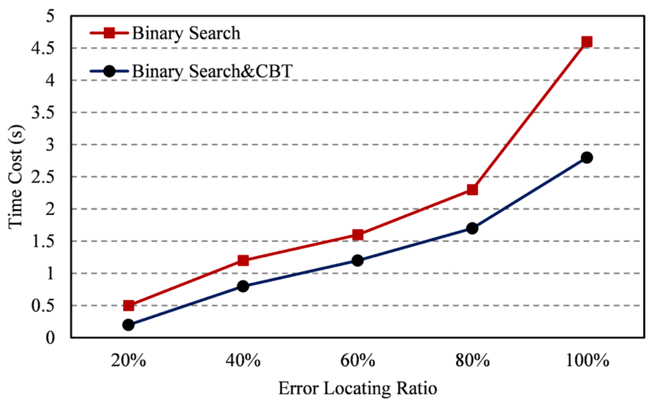

Fig. 5. The computational cost of the error locating ratio.

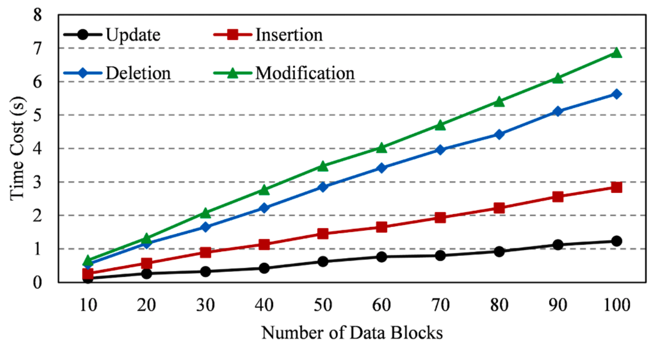

Fig. 6. The computational Cost of the data dynamics.

secure compared to traditional edge computing. Security analysis result demonstrates the correctness and detectability of the proposed scheme, and it can be performed with low false positive rate. Performance analysis result illustrates that our scheme needs less computational cost, which is practical for IoT in blockchain enabled edge computing.

CRediT authorship contribution statement

Dengzhi Liu: Writing – original draft. **Yong Zhang:** Supervision, Funding acquisition. **Dongbao Jia:** Formal analysis, Writing – review & editing. **Qiaosheng Zhang:** Methodology, Data curation. **Xuefeng Zhao:** Resources, Software. **Huan Rong:** Validation.

Declaration of Competing Interest

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

Acknowledgments

This work is supported by the Research Start-up Fund of JOU under Grant KQ20039, the Six Talent Peaks Project in Jiangsu Province under Grant XYDXX-140, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 19KJB160001, the Lianyungang city Haiyan project Grant 2019-QD-004, the North Jiangsu Science and Technology project under Grant SZ-LYG202024.

References

[1] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, N. Ghani, Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoTexploitations, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2702–2733, https://doi.org/10.1109/COMST.2019.2910750.
[2] B. Yin, X. Wei, Communication-efficient data aggregation tree construction for complex queries in IoT applications, IEEE Internet Things J. 6 (2) (2019) 3352–3363, https://doi.org/10.1109/JIOT.2018.2882820.
[3] C. Ge, Z. Liu, J. Xia, L. Fang, Revocable identity-based broadcast proxy re-encryption for data sharing in clouds, IEEE Trans. Dependable Secure Comput. 18 (3) (2021) 1214–1226, https://doi.org/10.1109/TDSC.2019.2899300.
[4] H. Ning, H. Liu, L.T. Yang, Aggregated-proof based hierarchical authentication scheme for the internet of things, IEEE Trans. Parallel Distrib. Syst. 26 (3) (2015) 657–667, https://doi.org/10.1109/TPDS.2014.2311791.
[5] C. Ge, W. Susilo, Z. Liu, J. Xia, F. Li, P. Szalachowski, Secure keyword search and data sharing mechanism for cloud computing, IEEE Trans. Dependable Secure Comput. (2020), https://doi.org/10.1109/TDSC.2020.2963978.
[6] Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, Y. Yang, A game theoretical approach for user allocation in edge computing environment, IEEE Trans. Parallel Distrib. Syst. 31 (3) (2020) 515–529, https://doi.org/10.1109/TPDS.2019.2938944.
[7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multiaccess edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1657–1681, https://doi.org/10.1109/COMST.2017.2705720.
[8] J. Ren, G. Yu, Y. He, G.Y. Li, Collaborative cloud and edge computing for latency minimization, IEEE Trans. Veh. Technol. 68 (5) (2019) 5031–5044, https://doi.org/10.1109/TVT.2019.2904244.
[9] Y. Ren, F. Zhu, S.P. Kumar, T. Wang, J. Wang, O. Alfarraj, A. Tolba, Data query mechanism based on hash computing power of blockchain in internet of things, Sensors. doi:10.3390/s20010207.
[10] T.T.A. Dinh, R. Liu, M. Zhang, G. Chen, B.C. Ooi, J. Wang, Untangling blockchain: a data processing view of blockchain systems, IEEE Trans. Knowl. Data Eng. 30 (7) (2018) 1366–1385, https://doi.org/10.1109/TKDE.2017.2781227.
[11] M. Belotti, N. Bozic, G. Pujolle, S. Secci, A vademecum on blockchain technologies: when, which, and how, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3796–3838, https://doi.org/10.1109/COMST.2019.2928178.
[12] R. Yang, F.R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: asurvey, some research issues and challenges, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1508–1532, https://doi.org/10.1109/COMST.2019.2894727.
[13] C. Ge, Z. Liu, L. Fang, H. Ling, A. Zhang, C. Yin, A hybrid fuzzy convolutional neural network based mechanism for photovoltaic cell defect detection with electroluminescence images, IEEE Trans. Parallel Distrib. Systems, 32 (7) (2021) 1653–1664, https://doi.org/10.1109/TPDS.2020.3046018.
[14] Y. Ren, Y. Leng, Y. Cheng, J. Wang, Secure data storage based on blockchain and coding in edge computing, Math. Biosci. Eng. 16 (4) (2019) 1874–1892, https://doi.org/10.3934/mbe.2019091.
[15] H. Liu, Y. Zhang, T. Yang, Blockchain-enabled security in electric vehicles cloud and edge computing, IEEE Netw. 32 (3) (2018) 78–83, https://doi.org/10.1109/MNET.2018.1700344.
[16] P. Zhang, H. Liu, Y. Zhang, Blockchain enabled cooperative authentication with data traceability in vehicular edge computing, Proceeding of the 2019 Computing, Communications and IoT Applications (ComComAp) (2019) 299–304. doi:10.1109/ComComAp46287.2019.9018754.
[17] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, W. Dou, Become: blockchain enabled computation offloading for IoT in mobile edge computing, IEEE Trans. Ind. Inf. 16 (6) (2020) 4187–4195, https://doi.org/10.1109/TII.2019.2936869.
[18] Y. Liu, F.R. Yu, X. Li, H. Ji, V.C.M. Leung, Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing, IEEE Trans. Veh. Technol. 68 (11) (2019) 11169–11185, https://doi.org/10.1109/TVT.2019.2937351.
[19] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, L. Fang, Revocable attribute-based encryption with data integrity in clouds, IEEE Trans. Dependable Secure Comput. (2021), https://doi.org/10.1109/TDSC.2021.3065999.
[20] C. Wang, Q. Wang, K. Ren, N. Cao, W. Lou, Toward secure and dependable storage services in cloud computing, IEEE Trans. Serv. Comput. 5 (2) (2012) 220–232, https://doi.org/10.1109/TSC.2011.24.
[21] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for secure cloud storage, IEEE Trans. Comput. 62 (2) (2013) 362–375, https://doi.org/10.1109/TC.2011.245.
[22] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, Enabling public auditability and data dynamics for storage security in cloud computing, IEEE Trans. Parallel Distrib. Syst. 22 (5) (2011) 847–859, https://doi.org/10.1109/TPDS.2010.183.
[23] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, Z. Han, When mobile blockchain meets edge computing, IEEE Commun. Mag. 56 (8) (2018) 33–39, https://doi.org/10.1109/MCOM.2018.1701095.
[24] R. Xu, Z. Zhang, B. Shen, Y. Zeng, C. Dai, Security storage based on fountain code and XORencryption in edge computing. Proceedings of the International Conference on Ubi-Media Computing (Ubi-Media), 2019, pp. 7–11, https://doi.org/10.1109/Ubi-Media.2019.00011.
[25] X. Zhang, R. Li, B. Cui, A security architecture of VANET based on blockchain and mobile edge computing. Proceedings of the International Conference on Hot Information-Centric Networking (HotICN), 2018, pp. 258–259, https://doi.org/10.1109/HOTICN.2018.8605952.
[26] J. Fu, Y. Liu, H. Chao, B.K. Bhargava, Z. Zhang, Secure data storage and searching for industrial IoTby integrating fog computing and cloud computing, IEEE Trans. Ind. Inform. 14 (10) (2018) 4519–4528, https://doi.org/10.1109/TII.2018.2793350.
[27] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, Provable data possession at untrusted stores. Proceedings of the ACM Conference on Computer and Communications Security, 2007, pp. 598–609, https://doi.org/10.1145/1315245.1315318.
[28] H. Shacham, B. Waters, Compact proofs of retrievability. Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, 2008, pp. 90–107, https://doi.org/10.1007/978-3-540-89255-7_7.
[29] B. Dan, X. Boyen, H. Shacham, Short group signatures. Proceedings of the International Cryptology Conference, 2004, pp. 41–55, https://doi.org/10.1007/978-3-540-28628-8_3.
[30] B.H. Bloom, Blockchain-enabled security in electric vehicles cloud and edge computing, Commun. ACM 13 (7) (1970) 422–426, https://doi.org/10.1109/MNET.2018.1700344.
[31] L. Fan, P. Cao, J. Almeida, A.Z. Broder, Summary cache: a scalable wide area web cache sharing protocol, IEEE/ACM Trans. Netw. 8 (3) (2000) 281–293, https://doi.org/10.1145/285237.285287.
[32] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, L. Fang, A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds, IEEE Trans. Dependable Secure Comput. (2021), https://doi.org/10.1109/TDSC.2021.3076580.
[33] The paring based cryptography library, 2013, http://crypto.stanford.edu/pbc/.

[34] The GNU multiple precision arithmetic library, 2013http://gmplib.org/.

[35] M. Yi, L. Wang, J. Wei, Distributed data possession provable in cloud, Distrib. Parallel Databases 35 (1) (2017) 1–21, https://doi.org/10.1007/s10619-016-7190-9.

[36] A.F. Barsoum, M.A. Hasan, Provable multicopy dynamic data possession in cloud computing systems, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 485–497, https://doi.org/10.1109/TIFS.2014.2384391.

Dengzhi Liu received the M.E. degree and Ph.D degree from the School of Computer and Software, Nanjing University of Information Science and Technology, in 2017 and 2020, respectively. He is currently an Assistant Professor with the School of Computer Engineering, Jiangsu Ocean University, China. He mainly focuses on the security and privacy issues in data storage and transmission. He has authored more than 50 research papers and published in international conferences and journals. His current research interests include cloud computing security, cyber security, and data security.

Yong Zhang received his M.E. degree in computer science from the School of Computer Science and Technology, Soochow University, China, in 2007. Now, he is working for his Ph.D. degree in China University of Mining and Technology. He is an associate professor at School of Computer Engineering, Jiangsu Ocean University, Lianyungang, China. He is a special expert on the development of Northern Jiangsu Province. His current research interests are theory of fresh E-commerce, data security, smart agriculture and intelligent information processing.

Dongbao Jia is an associate professor at School of Computer Engineering, Jiangsu Ocean University, Lianyungang, China. And he is also currently a special researcher at the University of Toyama, Toyama, Japan. He received his M.E. and Ph.D. degrees from the Department of Intellectual Information Engineering, Graduate School of Science and Engineering for Education, University of Toyama, Toyama, Japan, in 2015 and 2018, respectively. His main research interests are intelligence algorithm, signal processing and neural engineering.

Qiaosheng Zhang graduated from the computer department of Harbin Institute of Technology in 2019 and has a Ph.D. degree in engineering. He is an associate professor at School of Computer Engineering, Jiangsu Ocean University, Lianyungang, China. His primary research interests include big data analysis, network security and intelligent information processing.

Xuefeng Zhao received his M.S. degree in computer application technology from College of Information Engineering, Yangzhou University, China, in 2007 and the Ph.D. degree in mechanical and electronic engineering from the School of Manufacturing Science and Engineering, Sichuan University, China, in 2011, respectively. He is an associate professor at School of Computer Engineering, Jiangsu Ocean University, Lianyungang, China. His current research interests are machine learning, pattern recognition, network and data security.

Huan Rong received the Ph.D. degree in computer science from the Nanjing University of Information Science and Technology, Nanjing, China. He is a visiting scholar with the University of Central Arkansas. His research interests lie in deep learning and the application of artificial intelligence, especially on the sentiment analysis and other interdisciplinary task. His research contributions have been published in the Information Sciences, the IEEE Transactions on Affective Computing, the Soft Computing, etc. And now, he is a lecturer with the School of Artificial Intelligence, Nanjing University of Information Science & Technology.