scientific reports

OPEN

Data storage mechanism of industrial IoT based on LRC sharding blockchain

Yongjun Ren[1], Xinyu Liu[1], Pradip Kumar Sharma[2], Osama Alfarraj[3], Amr Tolba[3], Shenqing Wang[4] & Jin Wang[5]✉

With the rapid development of Industry 4.0, the data security of Industrial Internet of Things in the Industry 4.0 environment has received widespread attention. Blockchain has the characteristics of decentralization and tamper-proof. Therefore, it has a natural advantage in solving the data security problem of Industrial Internet of Things. However, current blockchain technologies face challenges in providing consistency, scalability and data security at the same time in Industrial Internet of Things. To address the scalability problem and data security problem of Industrial Internet of Things, this paper constructs a highly scalable data storage mechanism for Industrial Internet of Things based on coded sharding blockchain. The mechanism uses coded sharding technology for data processing to improve the fault tolerance and storage load of the blockchain to solve the scalability problem. Then a cryptographic accumulator-based data storage scheme is designed which connects the cryptographic accumulator with the sharding nodes to save storage overhead and solve the security problem of data storage and verification. Finally, the scheme is proved to be security and the performance of the scheme is evaluated.

Industrial Internet of Things (IIoT) is the continuous integration of various types of acquisition and control sensors or controllers with sensing and monitoring capabilities, as well as mobile communication, intelligent analysis and other technologies into various aspects of the industrial production process, thereby significantly improving manufacturing efficiency, improving product quality, reducing product costs and resource consumption, and ultimately achieving a new stage of upgrading traditional industry to intelligence[1,2]. IIoT as an emerging product, the architecture is more complex, there is no unified standard, all aspects of security problems are more prominent[3,4].

Among them, the data security of the IIoT is facing great challenges. Blockchain has the characteristics of decentralization and tamper-proof modification, and has natural advantages in solving the data security problems of the IIoT. The blockchain is layered through a peer-to-peer (P2P) network so that the whole network can perform complete information transmission and verify its accuracy[5–9]. In addition, the blockchain uses automatic filtering mode to establish credit information. This reliable information can effectively improve the security of IIoT transactions. More importantly, blockchain nodes can participate or leave independently without any interference to the whole blockchain[10–13]. Therefore, the blockchain solution can reasonably integrate networking data resources and improve the security of IIoT users.

However, because the current blockchain can't meet the dilemma of data security and scalability of IIoT, and the blockchain[14–17] itself still has security problems of data storage and verification, this paper integrates the coded sharding blockchain into IIoT. Therefore, the specific contributions of this paper are as follows:

(1) This paper studies the requirements of IIoT and analyzes the challenge of ensuring the scalability and data security of IIoT system while providing consistency with blockchain.

[1]Engineering Research Center of Digital Forensics, Ministry of Education, School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China. [2]Department of Computing Science, University of Aberdeen, Aberdeen AB24 3FX, UK. [3]Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia. [4]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. [5]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China. ✉email: jinwang@csust.edu.cn

(2) According to the requirements of system scalability and data security of IIoT, an IIoT storage architecture based on local repairable code (LRC) sharding blockchain is constructed by combining coded sharding blockchain with IIoT.
(3) This paper shows the scheme of connecting the bilinear accumulator with the shard node, using the accumulator as the storage structure for data storage. The functions of membership witness and membership verification of the accumulator can solve the data storage and verification security problems existing in the blockchain itself.

Results

System structure.

The IIoT network system shown in Fig. 1, which uses a sharding blockchain and consists of two main elements:

Global network. Global/centralized networks can provide the highest resource capacity. It follows the traditional centralized cloud computing method, consists of cloud servers that host software, is responsible for planning, monitoring and managing resources, and handles the logical execution of architectural functional components, and it provides a globally available service platform for applications that require high storage and computing power[18–20]. As the core architecture of IIoT, it is connected with the blockchain to form the overall system structure of the combination of IIoT and blockchain.

Edge network. The blockchain is connected through a global network as an edge network[21–23]. Each blockchain contains n shards, and each shard contains an edge node and an IIoT node that can be stored in the licensed blockchain, and the two nodes are connected to form a shard node. Finally, the edge network is connected with the global network to form the overall structure shown in Fig. 1. In order to further improve the system structure formed by IIoT and the sharding blockchain, we add the LRC to solve the IIoT data reliability problems, and add bilinear accumulator to solve the data storage and verification security problems of the blockchain itself. The structure realizes data storage security through bilinear accumulator, and uses coded sharding technology to ensure data reliability and blockchain scalability in the process of operation.

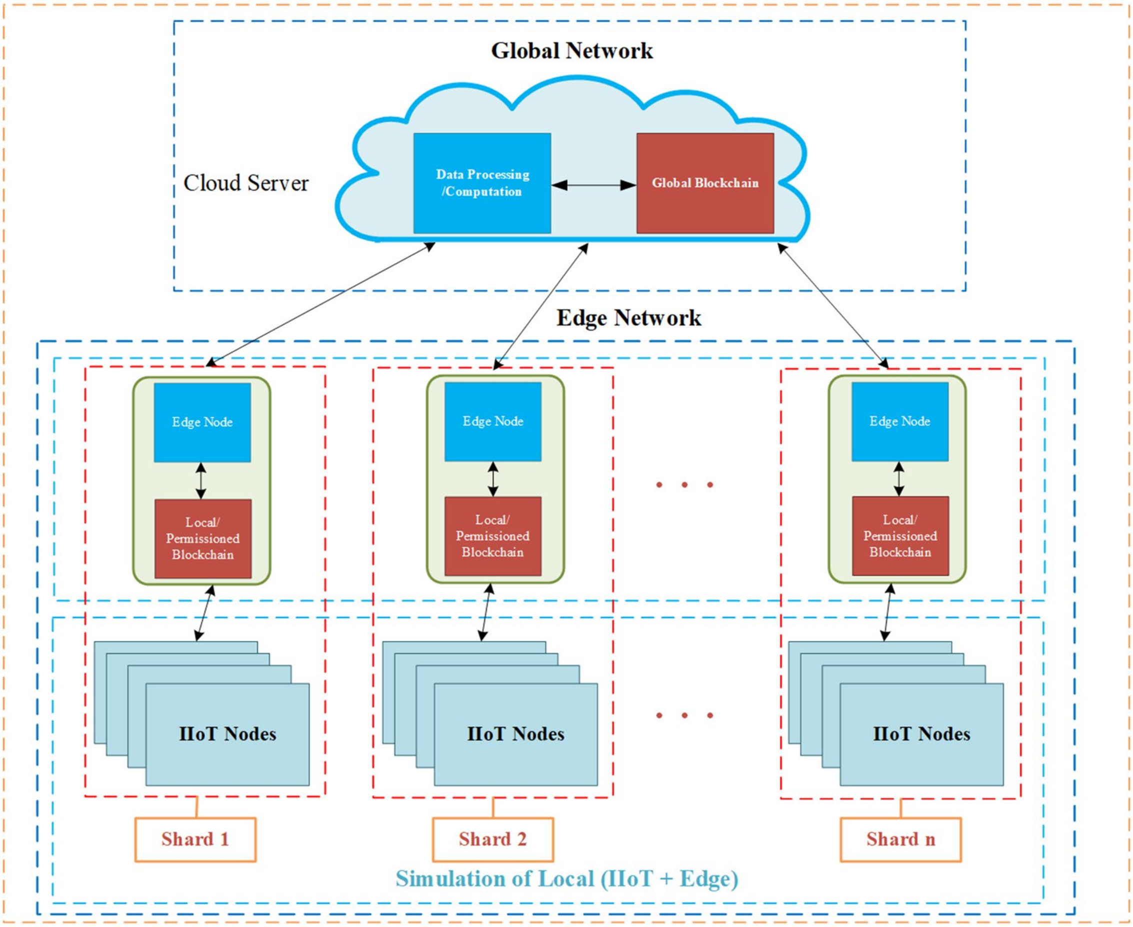
Global Network
Cloud Server
Data Processing /Computation
Global Blockchain
Edge Network
Edge Node
Local/ Permissioned Blockchain
Edge Node
Local/ Permissioned Blockchain
Edge Node
Local/ Permissioned Blockchain
IIoT Nodes
IIoT Nodes
IIoT Nodes
Shard 1
Shard 2
Shard n
Simulation of Local (IIoT + Edge)

Figure 1. Overall system structure diagram.

In the system structure, the coded sharding technology ensures the data reliability of IIoT and increases the fault tolerance of the blockchain system. At the same time, the anti-attack ability of the system can be increased by encoding the data.

LRC sharding blockchain structure.

In order to better ensure the data security of IIoT, we propose to encode the data blocks in each shard node of the blockchain. This kind of sharding coded is used to ensure the reliability of the data.

The LRC can not only ensure the reliability of the data in the blockchain, but also repair the wrong node data, which has lower bandwidth overhead and higher repair efficiency. Therefore, we choose the LRC to encode the node data blocks after the blockchain is sharded[24]. LRC sharding blockchain (LRC-SB) structure as shown in Fig. 2.

Discussion

No blockchain system can achieve consistency, security, and performance scalability at the same time. For the current blockchain system, as more nodes join the network, the efficiency of the system remains constant at most. One of the main ways to achieve blockchain scalability is to use the concept of shard[10,25,26] : It divides the network into multiple regions, which process the corresponding transactions in parallel. In the underlying public blockchain system, the transactions on the blockchain will be divided into different pieces, which are composed of different nodes on the network. Therefore, only a small part of the input transactions needs to be processed, and a lot of verification work can be done by processing in parallel with other nodes on the network. Dividing the network into fragments allows more transactions to be processed and verified at the same time, so in the blockchain, a single blockchain can be split into multiple sub-chains running in parallel, and different sub-chains deal with different parts of the blockchain, thus reducing the load on each individual node. However, the existing distribution proposal may damage the nodes in a given shard and cause permanent damage to the corresponding part, so as to achieve efficient scaling. In[27], the "polynomially coded sharding" scheme is proposed to realize the scalability and trust of the blockchain system. Although this method can also solve the security problems caused by shard. However, the security problem of data storage and verification in the blockchain itself has not been solved. Compared with the existing scheme[28], the bilinear mapping accumulator is faster than the RSA accumulator and performs better than expected in most cases where accumulators are needed. Therefore, this paper uses bilinear accumulator to solve the security problem of data storage and verification of blockchain.

A large amount of data needs to be stored and processed in IIoT. In order to improve the security and scalability of the current IIoT, this paper proposes to use local repairable code (LRC) sharding technology, combined with bilinear accumulator to solve this dilemma. Compared with the polynomially coded sharding technology, LRC sharding technology has better scalability and efficiency, because the data repair can be completed in the local network, reducing the communication cost of data repair. Therefore, we use local repair code (LRC) sharding technology to replace the "polynomially coding sharding" used in[27] to ensure the scalability of the system, but the security problems of data storage and verification in the blockchain itself have not been solved, so we

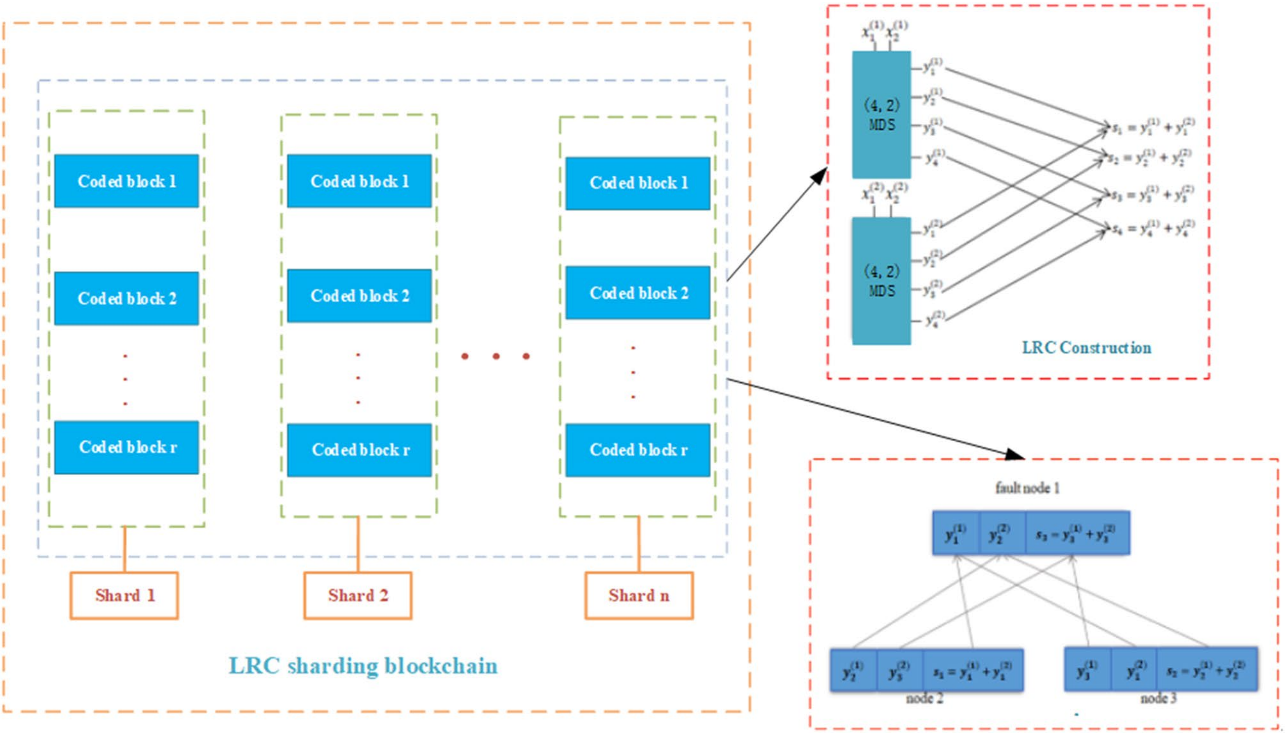

Figure 2. LRC sharding blockchain structure.

use bilinear accumulators to improve data security. Scalability essentially refers to the throughput of the system, in which the best way to improve throughput is shard technology. The main idea is to divide the blockchain into several independent sub-chains, and then process the data on each sub-chain separately. However, it is easy to have data error or node data loss, so we combine the LRC and shard technology, which have great advantages in repairing the error node data, to improve the throughput of the system on the basis of ensuring data integrity. Although the use of LRC[29–31] can ensure the integrity and correctness of data to a certain extent, it can't completely resist enemy attacks and ensure the security problems encountered in the process of data transmission. Therefore, we use the bilinear accumulator, which is widely used and efficient, in the sharding blockchain, connect the accumulator with the shard node, and use the accumulator as the storage structure for data storage. The functions of membership witness and membership verification of the accumulator are used to ensure the data security of the blockchain.

This paper proposes a data storage mechanism of IIoT based on LRC sharding blockchain, which not only ensures the reliability of IIoT data, but also increases the fault tolerance and anti-attack ability of the blockchain system. However, the operation process of adding a cryptographic accumulator to the blockchain is relatively complex to store data directly in the blockchain. In this process, we need to further verify the data, it can better ensure the security of blockchain data storage and verification, so how to simplify the operation flow of the system structure needs to be further discussed.

Methods

LRC construction.

Let the node x with the size of $M = rk$ data be divided into r parts, $x = \left[x^{(1)}, \ldots, x^{(r)}\right]$, where each $x^{(i)}, i \in [r]$, and the size is k. We use the extremal (n, k, d)-MDS (maximum distance separable code) code $y^{(1)} = x^{(1)}G, \ldots, y^{(r)} = x^{(r)}G$, where G is the $n \times k$ MDS generating matrix, and each part of the r part is independently encoded into a coding vector $y^{(i)}$ of length n, where $(r + 1) \mid n$.

As a MDS precoding[32], we use (n, k)-RS (Reed–solomon) code, which requires that every element of k is over a finite field F_{2^p}, such that $2^p \geq n$ for any p. This will mean that all the sub data stored in our coding is on finite field with size $2^p \geq n$. Afterwards, a single parity XOR vector $S = \bigoplus_{i=1}^{r} y^{(i)}$ is generated from the coding vectors.

The precoding process produces a total $r.n$ coding blocks, $y^{(i)}$ vectors and n XOR parity blocks in s vectors. Means of the total community $(r + 1).n$ blocks can be placed in n nodes, so we decide that each node stores $r + 1$ block. Therefore, each node needs to have the following storage capacity$\alpha = \frac{M}{k} + \frac{1}{r}\frac{M}{k} = r + 1(\text{ coded blocks })$.

According to the construction of the LRC, we can conclude that the specific algorithms for LRC to encode and decode the data in the shard are shown in Algorithms 1 and 2.

Algorithm 1 Coding the data in a shard.

```
1. Input $x^{(1)}, \ldots, x^{(r)}$ which are $r$ input blocks.
2. Use an outer $(n,k,d)$ MDS code to encode the $r$ input block to the encoding vector $y^{(1)}, \ldots, y^{(r)}$, where $y^{(1)} = x^{(1)}G, \ldots, y^{(r)} = x^{(1)}G, G$ is an $n \times k$ MDS generated matrix.
3. A single parity XOR vector $s$ is generated by coding vector $y^{(1)}, \ldots, y^{(r)}$.
4. In the above coding process, $r.n$ coding blocks are generated, which are marked as $C^{(1)}, C^{(2)}, \ldots, C^{(r \cdot n)}$.
5. Replace the input block $x^{(1)}, \ldots, x^{(r)}$ with the encoding block $C^{(1)}, C^{(2)}, \ldots, C^{(r \cdot n)}$.
```

Algorithm 2 LRC peeling decoding algorithm.

```
1. Input: all (r + 1)n coding blocks and $y^{(i)}$ vectors generated during the encoding process.
2. Initial check: check all zero-degree parity equations (that is, those whose coding symbols are known). If the parity equation fails, report the error coding proof and exit.
3. While not all M data symbols are restored do
4.     Find a parity equation of degree 1 with only one unknown coding symbol,
5.     Restore this encoded symbol and use its vector for verification. If it fails, report the error coding certificate and exit,
6.     Check all the relevant zero-degree parity equations. If any parity equation fails, report the error coding proof and exit.
7. end
```

LRC fault shard node repair.

To repair each missing node, it is necessary to contact r nodes, that is, the existence of locality is the code of r. In the case of general non-lost cases, the repair of nodes in the first repair group in $r + 1$ node needs to be considered. It is enough, because the node follows the put attribute of the same in different repair group.

The main observation is that each node stores a block with a different index of $r + 1$ in the repair group: the $r + 1$ blocks of a specific index are stored in $r + 1$ different nodes within a single repair group. Such as, when the first node is faulty, you need to download s_1 in the second node, download $y_1^{(r+1)}$ to regenerate the symbol $y_1^{(1)}$ of first row in the third node, and the rest may be deduced by analogy. Once all symbols are downloaded, $y_1^{(1)}$ is just a simple XOR from these symbols. By the same manner, when reconstructing the node in each repair group, first download the remaining identical index block, and then XOR them together, and finally regenerate the required lost blocks. Because of the reconstruction of each block needs to contact r other blocks, and can only be repaired in a single repair group with r remaining nodes, therefore the encoding has locality r.

Data storage and verification of LRC-SB based on bilinear accumulator.

The shard technology improves the throughput and scalability of the blockchain system, and LRC improves the data reliability of the blockchain. In the coded sharding blockchain, LRC is used to encode the data in the IIoT node to form redundant data, and the data is stored in the whole node of the corresponding shard of the edge network. However, the traditional blockchain must verify the whole node of the stored data and store all the data and membership witnesses in the node together, so the cost of running the full node and independent verification blockchain is very high, and there is no guarantee that the validated data is available and there is no hidden malicious data. Therefore, the verification efficiency of the original blockchain is very low, the node storage pressure is also very high, and cannot guarantee the availability of data[33,34]. Therefore, on the basis of LRC-SB, we propose to use pairing-based accumulator to solve these problems.

Construction of bilinear accumulator. The cryptographic accumulator scheme allows the finite set $X = \{x_1, \ldots, x_n\}$ to be accumulated into the accumulative value acc_X, the so-called accumulator. For each element $x_i \in X$, the so-called witness wit_{x_i} can be effectively calculated to prove membership of x_i in acc_X. Based on the $n -$ SDH assumption of the bilinear pair accumulator[35–37], works as follows:

(1) $\mathrm{Gen}\left(1^k, t\right)$: Input the security parameter k and select the three groups G_1, G_2 and G_T with prime order p to generate bilinear pairs $e : G_1 \times G_2 \to G_T$. Among them, the generator of G_1, G_2 is g_1, g_2, and $s \leftarrow Z_p^*$ is selected at the same time. Finally, the algorithm returns a set of keys $\left(sk_{acc}, pk_{acc}\right) \leftarrow \left(s, \left(g_1, g_1^s, \ldots, g_1^{s^t}\right)\right)$.

(2) $\mathrm{Eval}_r\left(\left(sk_{acc}, pk_{acc}\right), X\right)$: Input the key pair $\left(sk_{acc}, pk_{acc}\right)$ and the set $X = \{x_1, \ldots, x_n\}$. If sk_{acc} exists, the accumulative value of set X is calculated as follows: $acc_X = g_1 \prod_{i=1}^{n} (x_i + s)$. If sk_{acc} is unknown, the accumulative value of set X cannot be calculated directly. So, if $\prod_{x \in X} (x + s)$ is calculated and expressed as $\sum_{i=0}^{n} a_i \cdot s^i$, the accumulative value of set X can be calculated as follows: $\mathrm{acc}_X = \prod_{i=0}^{n} \left(g_1 s^i\right)^{a_i}$. Finally, the algorithm returns the accumulative value acc c_X and the auxiliary value $aux = (X)$.

(3) $\mathrm{WitCreate}\left(\left(sk_{acc}, pk_{acc}\right), acc_X, aux, x_i\right)$: Input the key pair $\left(sk_{acc}, pk_{acc}\right)$, accumulative value acc_X, auxiliary value $aux = (X)$ and element x_i. The algorithm first verifies whether element x_i belongs to set X, If not, return $\perp$. If sk_{acc} exists, the witness of element x_i can be calculated as follows: $\mathrm{wit}_{\,x_i} = acc_X^{(x_i+s)^{-1}}$. If the sk_{acc} is unknown, the wit_{x_i} cannot be calculated directly. But we can calculate $\prod_{x \in X \backslash x_i} (x + s)$ and represent it as $\sum_{i=0}^{n-1} a_i \cdot s^i$, then the witness of element x_i can be calculated as follows: $\mathrm{wit}_{\,x_i} = \prod_{i=0}^{n-1} \left(g_1^{s^i}\right)^{a_i}$.

(4) $\mathrm{Verify}\left(pk_{acc}, \mathrm{acc}_X, \mathrm{wit}_{\,x_i}, x_i\right)$: The algorithm determines whether element x_i is a member of set X by verifying that $e\left(\mathrm{acc}_X, g_2\right) = e\left(\mathrm{wit}_{\,x_i}, g_2 x_i g_2^s\right)$. If the above formula is true, the algorithm returns true, otherwise it returns false.

Storage and verification scheme description. As shown in Fig. 3, each shard constructs an accumulator, and the blocks in each shard are connected to the accumulator, and at the same time, each block contains a new accumulator to compress and store the encoded available data, and then validate the data to ensure that the data stored in the block is available.

All the shard full nodes are connected to an accumulator, and there is also an accumulator in the block of each shard node for data storage and data verification[38], it only needs to store the data that is considered to have data availability after verification. Because the encoded data is already resistant to attack, it is difficult for malicious nodes to hide some data in the encoded data, so it is difficult for attackers to hide some data in the shard nodes, then use bilinear accumulators to provide membership witness (CodeMemWitCreate) for the encoded data, and use VerifyCodeMem algorithm to authenticate the encoded data to ensure the availability of these data.

The whole node after shard can use the BatchAdd algorithm to add the encoded data with data availability verified to the node, and for the unencoded data or the discovered malicious hidden data, if it has been added to the shard node, it can be deleted using the BatchDel algorithm. The NonCodeMemWitCreate algorithm is used to provide non-membership witness for the unencoded data or the discovered malicious hidden data, and the VerifyCodeNonMem algorithm is used for non-membership witness. All the algorithms mentioned above are shown in Algorithm 3 and the parameters required for the above algorithm are shown in Table 1.

The bilinear accumulator is added to the encoded blockchain, so that the shard node only needs to store the encoded node and verify the encoded node, so it can reduce the data storage burden of the traditional blockchain to the node and improve the verification efficiency of the node, and then combined with coding to reduce the malicious hidden data of the attacker while providing membership witness for the encoded data. Further solve the problem of data availability of blockchain.

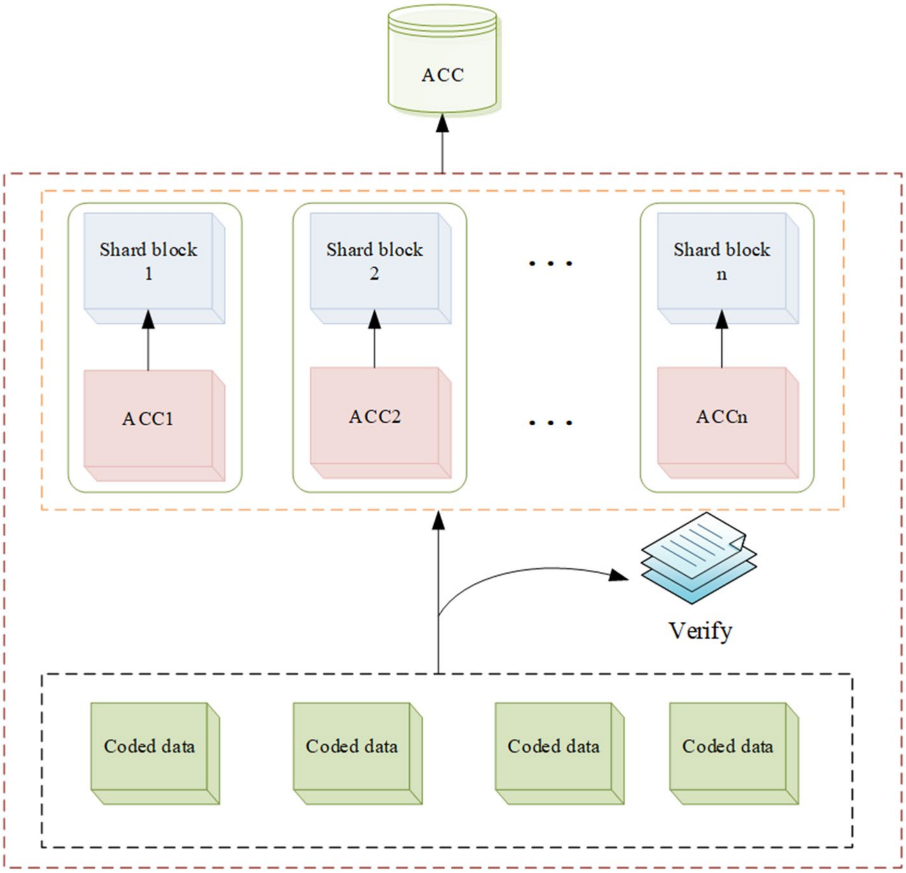

Figure 3. Coded sharding blockchain storage scheme based on bilinear accumulator.

λ	Security parameters
A_t	A discrete time counter
$D = \left\{d_{1,\ldots,}d_n\right\}$	Accumulator value at time t
D_0	Current accumulated elements set
m	A subset of the set D
w_x^t	Information for updating certificates
u_x^t	Membership witness
Bezout (x, y)	The subprocess of outputting the Bezout coefficient $a, b \in Z$ for a pair of integers x, y (that is, satisfying the relation $ax + by = 1$) which are primes to each other

Table 1. Algorithm parameters.

Algorithm 3 Accumulator related algorithm.

```
(1) Parameter generation algorithm
Gen($\lambda$) :
$G_1, G_2 \leftarrow G\,\mathrm{Gen}(\lambda)$
$g_1, g_2 \leftarrow G_1, G_2$
return $G_1, G_2, g_1, g_2$

(2) Element addition algorithm
Add$(A_t, D, x)$ :
if $x \in D$
return$A_t$
else $D \leftarrow D \cup \{x\}$
  $m \leftarrow x$
  $A_{t+1} \leftarrow A_t^x$
  return$A_{t+1}, m'$

(3) Batch addition algorithm
BatchAdd $(A_t, \{x_1, \ldots, x_n\})$ :
  $x^* \leftarrow \prod_{i=1}^n x_i$
  $A_{t+1} \leftarrow A_t^{x*}$
  return $A_{t+1}$, $\mathrm{NI} - \mathrm{PoE}\,(x^*, A_t, A_{t+1})$

(4) Element deletion algorithm
Del$(A_t, D, x)$ :
if $x \notin D$
return $A_t$
else $D \leftarrow D\{x\}$
$A_{t+1} \leftarrow g_{d \in D}(d+k)$
$m \leftarrow \{x, A_t, A_{t+1}\}$
return $A_{t+1}, m$

(5) Batch deletion algorithm
BatchDel$\left(A_t, \left(x_1, w_{x_1}^t\right), \ldots, \left(x_n, w_{x_n}^t\right)\right)$ :
$A_{t+1} \leftarrow w_{x_1}^t$
$x^* \leftarrow x_1$
for $i \leftarrow 2, i \leq n$
  $A_{t+1} \leftarrow w_{x_n}^t$
  $x^* \leftarrow x^* x_i$
return $A_{t+1}$, NI-PoE $(x^*, A_{t+1}, A_t)$

(6) Create a membership witness for
    encoded data
CodeMemWitCreate $(A_t, D, x)$ :
  $w_x^t \leftarrow g_{d \in D, d \neq x}(d+k)$
  return $w_x^t$

(7) Create a non-membership witness
    for non-encoded data
NonCodeMemWitCreate $(A_t, D, x)$ :
$d^* \leftarrow \prod_{d \in D}(d+k)$
$a, b \leftarrow \mathrm{Bezout}\,(d^*, x)$
else $D \leftarrow D\{x\}$
$u_x^t \leftarrow \left(a, g^b\right)$
return $u_x^t$

(8) Verify the membership of
    the encoded data
VerifyCodeMem $(A, w_x, x)$ :
if $e\left(w_x, g_2^{\prod_{d_0 \in D_0}(d_0+k)}\right) = e\,(A, g_2)$
return 1

(9) Verify non-membership of non-
    encoded data
VerifyNonCodeMem $(A, u_x, x)$ :
if $e\left(A^a g_1^{bx}, g_2\right) = e\,(g_1, g_2)$
return 1
```

Adversary model. The data of IIoT is stored on several untrusted nodes of the blockchain system. We consider an isomorphic synchronous network, that is, all nodes have similar processing power, and the communication delay between any pair of nodes is bounded by a known constant. Some nodes may be corrupted and affected by Byzantine failures, that is, they may calculate and transmit arbitrarily incorrect results during block verification. Our goal is to design a secure verification scheme for the following strong adversary models:

1. Attackers can destroy a fixed proportion of network nodes, that is, the number of malicious nodes increases linearly.
2. If a traditional sharding solution is used, the attacker knows the node to shard allocation and can adaptively select the node to attack.

We note that under this opponent model, the random sharding rotation method is no longer secure, because after knowing which nodes are assigned to the shard, the adversary can concentrate on attacking a single shard. Therefore, we use the coding sharding technology in the system model, which can better ensure the reliability of the data after the sharding is attacked, even if the encoded data is attacked, it is difficult to obtain the correct data information. Then the membership witness function of the bilinear accumulator is used to ensure the data verification security of the system.

In the data storage process of blockchain, the system is vulnerable to data availability attacks, and coded sharing technology can better ensure data reliability. Any small hiding on the original block caused by the encoded data will be tantamount to making a large part of the coded block unavailable, which can be detected by the node by exponentially increasing the probability of random sampling of the coded block.

Security analysis. The data storage and verification scheme of the coded sharding blockchain in IIoT is exposed to various attacks. We will show how we propose how to defend these possible attacks.

Data delete attack. If the shard blockchain is an edge network that deletes and destroys the original data, the witness of the encoded data block cannot be calculated: $w_x^t \leftarrow g_1^{\prod_{d \in D, d \neq x}(d+k)}$.

When it receives the verifier verification. Since each encoded data block in the sharding blockchain is a coded accumulated value A_t, the IIoT global network cannot only use original data to generate a valid ownership certificate.

Replacement attack. If the IIoT edge network replaces damaged or deleted data with other valid encoded data blocks, and when the verifier uses a random block witnessed $w_x^{'\mathrm{t}}$ to verify the encoded data block in the edge network of IIoT, the verifier is calculated witness $w_x^{'\mathrm{t}}$ cannot match the extracted member witness w_x^t.

Data hiding attack. The data owner uses the local repair code to encode the data in the node to generate encoded data block $C^{(1)}, C^{(2)}, \ldots, C^{(r\cdot n)}$. Because the encoded data is resistant to attack, it is difficult for malicious nodes to hide in the encoded data, so the possibility of attackers hiding some data in shard nodes is almost negligible.

Replay attack. It makes no sense for cloud service providers to cache witnesses from past computing to meet new challenges from current validators. On the one hand, the cloud service provider cache witness needs to store both the target block and the corresponding witness, which will greatly increase the storage overhead of the cloud service provider. On the other hand, the verifier challenges the cloud storage provider with the cumulative membership of random data blocks when verifying data integrity, so the probability of the same challenge is basically negligible.

Data disclosure attack. In the setup phase, the data owner accumulates each encoded data block using a bilinear accumulator, and the data accumulated in the accumulated set has its own membership proof. As a result, others cannot know the data outsourced to IIoT cloud service providers.

Double spending attack. All data needs to be compressed and stored through a bilinear accumulator, and membership verification is required in the stored process. Even if a 51% attack is launched to copy the same data for operation, membership verification is required. However, the data already stored in the cryptography accumulator will not be added again, so the purpose of the double spending attack will not be achieved.

Experimental and analysis

Performance evaluation.

We use Aliyun and Amazon S3. Compared with a single cloud storage service, the response time of a distributed cloud storage service composed of multiple cloud storage services is more stable and faster. This is because the cloud storage based on blockchain provides multiple copies of data and can take advantage of the network bandwidth of multiple cloud storage services to overcome the bandwidth shortage of a single cloud server.

In the case of replica repair based on blockchain, when local repair codes are used to repair faulty data, the repair bandwidth and disk I/O will be reduced. As the redundancy m increases, it helps to repair the decrease in bandwidth and disk I/O. In view of the fact that our system can significantly reduce the amount of storage required by nodes without significantly increasing CPU decryption, running the blockchain on IIoT devices is a very important step.

The central goal of the system is scalability. Let N be the number of nodes and b the block size. Consider the simplest extension solution, which is to spread data across the network without duplication. Storage overhead measures the ratio of the total storage cost to the actual storage information. Consider that everyone keeps a complete copy of the data and the storage overhead is O(N), which indicates that the storage cost increases linearly with the size of the network. LRC sharding blockchain realizes O(1) storage in real time on the client side, and O(Logb) storage in case of client damage. When using renewable code, the worst-case scenario is that the opponent sends the wrong block prediction node that needs to download O(b) data to prevent fraud. LRC sharding blockchain achieves near-optimal overhead and only needs to download O(Logb) proof.

We measure the amount of data stored by each node after 1500 blocks (about 6 million transactions) are processed in the LRC sharding blockchain. Compared to previous work, we estimated the storage space required for each node in PolyShard and RC-blockchain based on the reported throughput and the number of shards of similar network size, as shown in Table 2.

Performance analysis.

Bandwidth consumption during sharding node coding. In each shard, we can calculate the bandwidth consumed by the node in the coding process. We evaluate the bandwidth consumption of each node in which each shard node stores d encoding fragments. The bandwidth consumption of the nodes in the shard changes when coding at k=10, 20, 30, 40, 50, 60 and d = 4, 5, 6, where k represents the total number of coding fragments. Figure 4 shows the shard node storing code fragment in the process of encoding bandwidth consumption. It can be seen that when d is fixed, k is, the greater, the less bandwidth is consumed to store encoding fragments in the process of node coding. When k is fixed, d is larger, the greater the bandwidth consumption of storing encoded fragments in the coding process is. Therefore, the bandwidth occupied by the node for data transmission in the shard is related to the amount of coded data allocated to the node. When the block size is relatively large, the bandwidth consumption and the amount of data stored can be measured, and a better scheme can be chosen. However, with the current mainstream blockchain block size, the bandwidth occupied by data transmission between nodes in the packet is very small.

Protocol	Network size	Storage
PolyShard	1200 nodes	500 MB
RC blockchain	1200 nodes	700 MB
LRC-SB	1200 nodes	238 MB

Table 2. Storage required for each node after 5 million transactions have been processed.

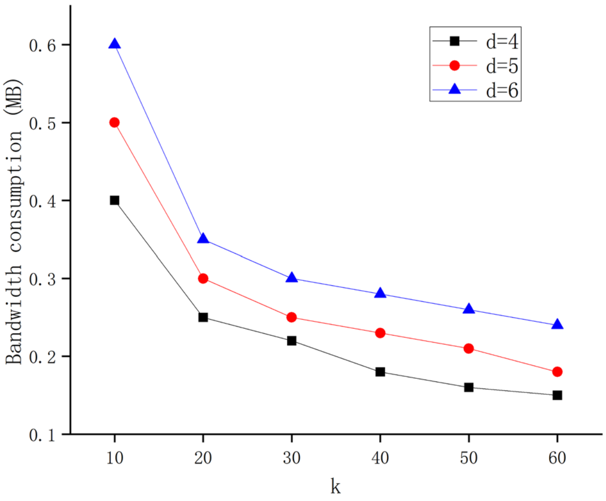

Figure 4. Bandwidth consumption when sharding nodes are encoded.

Error node repair rate of LRC. In the experiment, we calculated the repair rate of the error nodes in each shard and evaluated the repair rate of the error nodes in the number of error nodes respectively. The total amount of encoded data of the nodes in the slice is the repair rate of the error nodes. Figure 5 shows the change in the repair rate of the error node. We can know that when p is fixed, the larger n is, the slower the repair rate of error nodes in each shard is. When n is fixed, the smaller p is, the faster the repair rate of error nodes is.

Performance analysis of LRC-SB. In order to better demonstrate the performance advantages of LRC-SB, we compare it with unencoded blockchains and renewable code-based blockchains[39,40]. Table 3 shows in detail the efficiency comparison of LRC-SB, RC blockchain and traditional blockchain in terms of throughput, storage efficiency and security, so as to show the advantages of LRC sharding.

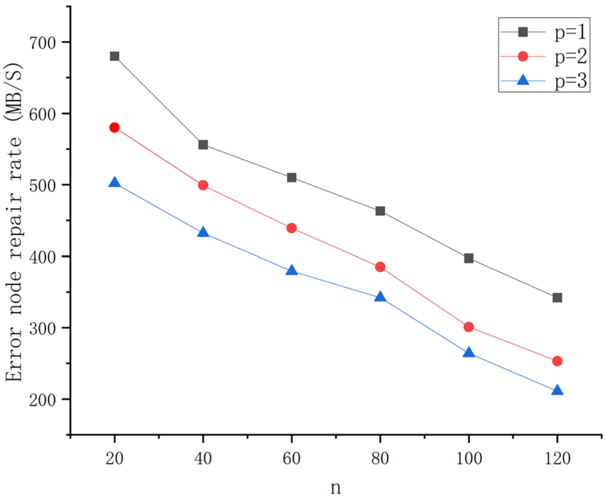

Figure 5. Error node repair rate.

Metrics	Blockchain	RC blockchain	LRC-SB
Throughput	$O(n)$	$O(n)$	$O(1)$
Storage efficient	$O\left(n^2\right)$	$O(n \log n)$	$O(n)$
Security	$O(n)$	$O(logn)$	$O(1)$

Table 3. Comparison of performance and efficiency of three kinds of blockchains.

In order to reduce the storage pressure of the node, the LRC-SB stores the encoded data with the help of an accumulator in the shard node, so the storage efficiency of the blockchain is specially tested. By comparing the storage efficiency of the traditional blockchain and the RC blockchain, we find that the storage efficiency of the LRC-SB is greatly improved after adding the accumulator. By comparing the changes in the reduction rate of storage overhead among the three, the result is shown in Fig. 6.

Analysis and comparison of system security.

In order to better demonstrate the performance advantages of LRC-SB, we compare it with unencoded blockchains and renewable code-based blockchains, and analyze the differences among them in terms of throughput, latency, scalability, fault tolerance and security. The results are shown in Table 4.

Conclusion

The Industrial IoT provides great convenience for industrial development through data exchange and integrated control. However, the data security problem of the IIoT still exists. Blockchain has great potential in dealing with data security issues of the IIoT. However, blockchain has certain scalability problems and cannot meet the scalability requirements of IIoT. To solve the problem, this paper uses shard technology to solve the scalability of IIoT system. And the local repairable code technology is utilized to encode the data in the sharding blockchain to enhance the reliability of the data. Moreover, this paper connects the accumulator with the shard node and uses the accumulator as a data storage structure. The function of membership witness and membership verification of the accumulator can ensure the data security of the blockchain and solve the security problems of data storage and verification in the blockchain itself. Meanwhile, the data reliability and availability of the blockchain are also improved by using LRC sharding technology and accumulators.

In the blockchain system, data security and privacy are always a tradeoff[41,42]. In order to solve the problem of data security and privacy of blockchain at the same time, we intend to improve the cryptographic accumulator in the future research work, and use the cryptographic accumulator with zero knowledge to realize the tradeoff between data security and privacy of blockchain.

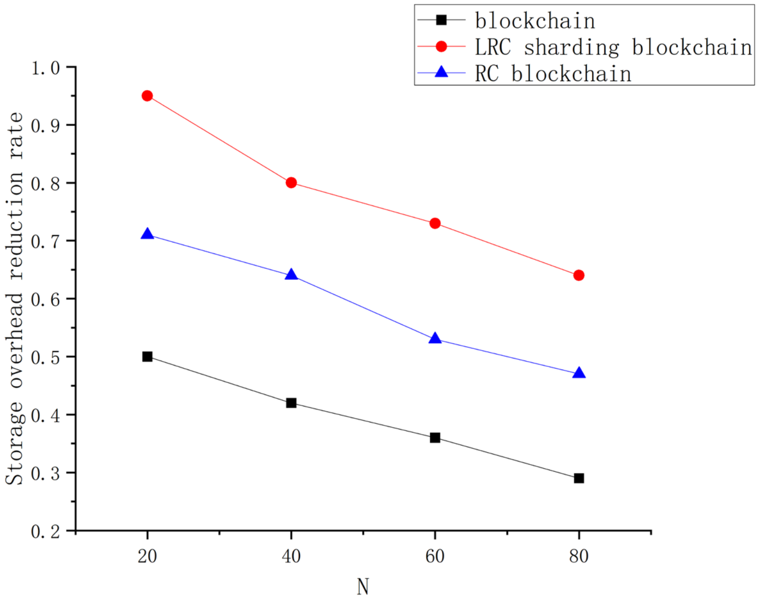

Figure 6. Reduction rate of storage overhead.

Metrics	Blockchain	RC blockchain	LRC-SB
Throughput	Around 1000 tps	Around 1000 tps	More than 1500 tps
Latency	Around 100 seconds	Around 100 seconds	Around 80 seconds
Scalability	Keep constant	Coding improves scalability	Coded sharding improves scalability
Fault tolerance	Unaffected	Affected by code repair rate	Affected by code repair rate
Security	Vulnerable	Anti-attack	Anti-attack

Table 4. Comparison of performance and efficiency of three kinds of blockchains.

Data availability

The datasets generated and analyzed during the current study are not publicly available due to restricted data sources but are available from the corresponding author on reasonable request.

Received: 5 November 2022; Accepted: 13 February 2023
Published online: 16 February 2023

References

1. Wang, J., Gao, Y., Zhou, C., Sherratt, S. & Wang, L. Optimal coverage multi-path scheduling scheme with multiple mobile sinks for wsns. *Comput. Mater. Contin.* **62**, 695–711. https://doi.org/10.32604/cmc.2020.08674 (2020).
2. Choo, K. R., Gritzalis, S. & Park, J. H. Cryptographic solutions for industrial internet-of-things: Research challenges and opportunities. *IEEE Trans. Ind. Inform.* **14**, 3567–3569. https://doi.org/10.1109/TII.2018.2841049 (2018).
3. Ren, Y., Zhu, F., Wang, J., Sharma, P. K. & Ghosh, U. Novel vote scheme for decision-making feedback based on blockchain in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **23**, 1639–1648. https://doi.org/10.1109/TITS.2021.3100103 (2022).
4. Serror, M., Hack, S., Henze, M., Schuba, M. & Wehrle, K. Challenges and opportunities in securing the industrial internet of things. *IEEE Trans. Ind. Inform.* **17**, 2985–2996. https://doi.org/10.1109/TII.2020.3023507 (2021).
5. Hui, H., Zhou, C., Xu, S. & Lin, F. A novel secure data transmission scheme in industrial internet of things. *China Commun.* **17**, 73–88. https://doi.org/10.23919/JCC.2020.01.006 (2020).
6. Wang, J., Han, C., Yu, X., Ren, Y. & Sherratt, S. Distributed secure storage scheme based on sharding blockchain. *Comput. Mater. Contin.* **70**, 4485–4502. https://doi.org/10.32604/cmc.2022.020648 (2022).
7. Ren, Y. *et al.* Multiple cloud storage mechanism based on blockchain in smart homes. *Future Generat. Comput. Syst.* **115**, 304–313. https://doi.org/10.1016/j.future.2020.09.019 (2021).
8. Sisinni, E., Saifullah, A., Han, S., Jennehag, U. & Gidlund, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Trans. Ind. Inform.* **14**, 4724–4734. https://doi.org/10.1109/TII.2018.2852491 (2018).
9. Laraib, J. *et al.* Sharechain: Blockchain-enabled model for sharing patient data using federated learning and differential privacy. *Expert Syst.* **18**, e13131. https://doi.org/10.1111/exsy.13131 (2022).
10. Yun, J., Goh, Y. & Chung, J.-M. Dqn-based optimization framework for secure sharded blockchain systems. *IEEE Internet Things J.* **8**, 708–722. https://doi.org/10.1109/JIOT.2020.3006896 (2021).
11. Liu, M., Yu, F. R., Teng, Y., Leung, V. C. M. & Song, M. Performance optimization for blockchain-enabled industrial internet of things (iiot) systems: A deep reinforcement learning approach. *IEEE Trans. Ind. Inform.* **15**, 3559–3570. https://doi.org/10.1109/TII.2019.2897805 (2019).
12. Medhane, D. V., Sangaiah, A. K., Hossain, M. S., Muhammad, G. & Wang, J. Blockchain-enabled distributed security framework for next-generation iot: An edge cloud and software-defined network-integrated approach. *IEEE Internet Things J.* **7**, 6143–6149. https://doi.org/10.1109/JIOT.2020.2977196 (2020).
13. Ma, Z. *et al.* A blockchain-based trusted data management scheme in edge computing. *IEEE Trans. Ind. Inform.* **16**, 2013–2021. https://doi.org/10.1109/TII.2019.2933482 (2020).
14. Yu, Y., Li, Y., Tian, J. & Liu, J. Blockchain-based solutions to security and privacy issues in the internet of things. *IEEE Wirel. Commun.* **25**, 12–18. https://doi.org/10.1109/MWC.2017.1800116 (2018).
15. Gopalan, A., Sankararaman, A., Walid, A. & Vishwanath, S. Stability and scalability of blockchain systems. *Proc. ACM Meas. Anal. Comput. Syst.* **4**, 1–35. https://doi.org/10.1145/3392153 (2020).
16. Pan, C., Liu, Z., Liu, Z. & Long, Y. Research on scalability of blockchain technology: Problems and methods. *J. Comput. Res. Dev.* **55**, 2099–2110. https://doi.org/10.7544/issn1000-1239.2018.20180440 (2018).
17. Ayesha, A. *et al.* A survey of blockchain technology: Architecture, applied domains, platforms, and security threats. *Soc. Sci. Comput. Rev.*https://doi.org/10.1177/08944393221110148 *(2022).*
18. Ren, Y., Huang, D., Wang, W. & Yu, X. Bsmd:a blockchain-based secure storage mechanism for big spatio-temporal data. *Future Generat. Comput. Syst.* **138**, 328–338. https://doi.org/10.1016/j.future.2022.09.008 (2023).
19. Liu, C., Li, K. & Li, K. A game approach to multi-servers load balancing with load-dependent server availability consideration. *IEEE Trans. Cloud Comput.* **9**, 1–13. https://doi.org/10.1109/TCC.2018.2790404 (2021).
20. Musa, Y. B., Rabia, L., Aisha, Y., Iqbal, K. M. & Ibrahim, M. A. Ricechain: Secure and traceable rice supply chain framework using blockchain technology. *PeerJ Comput. Sci.* **8**, e801. https://doi.org/10.7717/peerj-cs.801 (2022).
21. Wang, J., Ju, C., Gao, Y., Sangaiah, A. K. & Kim, G. J. A pso based energy efficient coverage control algorithm for wireless sensor networks. *Comput. Mater. Contin.* **56**, 433–446. https://doi.org/10.3970/cmc.2018.04132 (2018).
22. Ren, Y., Leng, Y., Cheng, Y. & Wang, J. Secure data storage based on blockchain and coding in edge computing. *Math. Biosci. Eng.* **16**, 1874–1892. https://doi.org/10.3934/mbe.2019091 (2019).
23. Yakubu, B. M., Khan, M. I., Javaid, N. & Khan, A. Blockchain-based secure multi-resource trading model for smart marketplace. *Computing* **103**, 379–400. https://doi.org/10.1007/s00607-020-00886-7 (2021).
24. Kumar, S., Graell i Amat, A., Andriyanova, I., Brännström, F. & Rosnes, E. Code constructions for distributed storage with low repair bandwidth and low repair complexity. *IEEE Trans. Commun.* **66**, 5847–5860. https://doi.org/10.1109/TCOMM.2018.2858765 (2018).
25. Mizrahi, A. & Rottenstreich, O. Blockchain state sharding with space-aware representations. *IEEE Trans. Netw. Serv. Manag.* **18**, 1571–1583. https://doi.org/10.1109/TNSM.2020.3031355 (2021).
26. Huang, C. *et al.* Repchain: A reputation-based secure, fast, and high incentive blockchain system via sharding. *IEEE Internet Things J.* **8**, 4291–4304. https://doi.org/10.1109/JIOT.2020.3028449 (2021).
27. Li, S. *et al.* Polyshard: Coded sharding achieves linearly scaling efficiency and security simultaneously. *IEEE Trans. Inf. Forens. Secur.* **16**, 249–261. https://doi.org/10.1109/TIFS.2020.3009610 (2021).
28. Boneh, D., Büunz, B., Fisch, B. Batching. & techniques for accumulators with applications to iops and stateless blockchains. In Advances in Cryptology-CRYPTO,. 39th Annual International Cryptology Conference, Santa Barbara, CA. *USA* **561–586**, 2019. https://doi.org/10.1007/978-3-030-26948-7_20 (2019).
29. Luo, Y., Xing, C. & Yuan, C. Optimal locally repairable codes of distance 3 and 4 via cyclic codes. *IEEE Trans. Inf. Theory* **65**, 1048–1053. https://doi.org/10.1109/TIT.2018.2854717 (2019).
30. Huang, P., Yaakobi, E., Uchikawa, H. & Siegel, P. H. Binary linear locally repairable codes. *IEEE Trans. Inf. Theory* **62**, 6268–6283. https://doi.org/10.1109/TIT.2016.2605119 (2016).
31. Jin, L., Ma, L. & Xing, C. Construction of optimal locally repairable codes via automorphism groups of rational function fields. *IEEE Trans. Inf. Theory* **66**, 210–221. https://doi.org/10.1109/TIT.2019.2946637 (2020).
32. Papailiopoulos, D. S. & Dimakis, A. G. Locally repairable codes. *IEEE Trans. Inf. Theory* **60**, 5843–5855. https://doi.org/10.1109/TIT.2014.2325570 (2014).
33. Wang, J., Gao, Y., Liu, W., Sangaiah, A. K. & Kim, H.-J. An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **15**, 1–9. https://doi.org/10.1177/1550147719839581 (2019).

34. Liang, X. *et al.* 17th IEEE/ACM international symposium on cluster. *Cloud Grid Comput. (CCGRID)* **1**(468–477), 2017. https://doi.org/10.1109/CCGRID.2017.8 (2017).
35. Benaloh, J. & Mare, M. D. One-way accumulators: A decentralized alternative to digital signatures. In Workshop on the Theory and Application of of Cryptographic Techniques, 274–285, https://doi.org/10.1007/3-540-48285-7_24 (1993).
36. Fueyo, M. & Herranz, J. On the efficiency of revocation in rsa-based anonymous systems. *IEEE Trans. Inf. Forens. Secur.* **11**, 1771–1779. https://doi.org/10.1109/TIFS.2016.2559443 (2016).
37. Chen, H. C. & Lee, P. P. Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation. *IEEE Trans. Parallel Distrib. Syst.* **25**, 407–416. https://doi.org/10.1109/TPDS.2013.164 (2014).
38. Ren, Y. *et al.* Data query mechanism based on hash computing power of blockchain in internet of things. *Sensors* **20**, 207. https://doi.org/10.3390/s20010207 (2019).
39. Sarkar, M. N. I., Meegahapola, L. G. & Datta, M. Reactive power management in renewable rich power grids: A review of grid-codes, renewable generators, support devices, control strategies and optimization algorithms. *IEEE Access* **6**, 41458–41489. https://doi.org/10.1109/ACCESS.2018.2838563 (2018).
40. Wang, J., Gao, Y., Liu, W., Wu, W. & Lim, S.-J. An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks. *Comput. Mater. Contin.* **58**, 711–725. https://doi.org/10.32604/cmc.2019.05450 (2019).
41. Randhir, K. *et al.* A privacy-preserving-based secure framework using blockchain-enabled deep-learning in cooperative intelligent transport system. *IEEE Trans. Intell. Transp. Syst.* **23**, 16492–16503. https://doi.org/10.1109/TITS.2021.3098636 (2022).
42. Prabhat, K., Randhir, K. P., Rakesh, T. & Gautam, S. P2tif: A blockchain and deep learning framework for privacy-preserved threat intelligence in industrial iot. *IEEE Trans. Ind. Inform.* **18**, 6358–6367. https://doi.org/10.1109/TII.2022.3142030 (2022).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos. 62072249, 62072056, 62076125). This work is also funded by the National Science Foundation of Hunan Province (2022JJ2029), and Hunan Provincial Key Research and Development Program (Grant No. 2022GK2019). This work is also funded by the Researchers Supporting Project number (RSP2023R102) King Saud University, Riyadh, Saudi Arabia. This work was also supported by the National Key R & D Program of Guangdong Province (Grant No. 2020B0101090002), and the Natural Science Foundation of Jiangsu Province (Grant Nos. BK20200418, BE2020106).

Author contributions

R.Y. and L.X. were responsible for conceptual analysis, methodological analysis and writing the original draft. P.K., O.A., A.T. and W.S. were responsible for thesis revision and review. W.J. was responsible for review, supervision, and project administration. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.