Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.037532
Article

Lightweight Storage Framework for Blockchain-Enabled Internet of Things Under Cloud Computing

Xinyi Qing[1,3], Baopeng Ye[2], Yuanquan Shi[1,3], Tao Li[4,*], Yuling Chen[4] and Lei Liu[1]

[1]School of Computer and Artificial Intelligence, Huaihua College, Huaihua, 418000, China
[2]Guizhou Science and Technology Innovation Service Center Co, Guiyang, 550000, China
[3]Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, Huaihua, 418000, China
[4]State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, 550000, China
*Corresponding Author: Tao Li. Email: litao_2019@qfnu.edu.cn

Received: 07 November 2022; Accepted: 15 January 2023

Abstract: Due to its decentralized, tamper-proof, and trust-free characteristics, blockchain is used in the Internet of Things (IoT) to guarantee the reliability of data. However, some technical flaws in blockchain itself prevent the development of these applications, such as the issue with linearly growing storage capacity of blockchain systems. On the other hand, there is a lack of storage resources for sensor devices in IoT, and numerous sensor devices will generate massive data at ultra-high speed, which makes the storage problem of the IoT enabled by blockchain more prominent. There are various solutions to reduce the storage burden by modifying the blockchain's storage policy, but most of them do not consider the willingness of peers. In attempt to make the blockchain more compatible with the IoT, this paper proposes a storage optimization scheme that revisits the system data storage problem from a more practically oriented standpoint. Peers will only store transactional data that they are directly involved in. In addition, a transaction verification model is developed to enable peers to undertake transaction verification with the aid of cloud computing, and an incentive mechanism is premised on the storage optimization scheme to assure data integrity. The results of the simulation experiments demonstrate the proposed scheme's advantage in terms of storage and throughput.

Keywords: Blockchain; internet of things; storage optimization; transaction verification; cloud computing; incentive mechanism

1 Introduction

In its long-term development and evolution, IoT technology has greatly improved the intelligence of people's lives and made the connection between people and devices, as well as devices and devices, closer. IoT technologies are widely used in smart cities [1], transportation [2–4], climate prediction

[5], and industry [6]. As 5G technology becomes widespread, the number of IoT devices and sensors will continue to grow at an accelerated rate, which means there will be more than 41 billion IoT devices worldwide and generate up to 80 ZB of data by 2025 [7]. Due to the huge number of IoT devices, it is likely to face a series of problems such as data loss [8], insecure data sharing, privacy leakage [9], centralized configuration and deployment of management services, and a lack of trust among participants. At the same time, the cost control of IoT devices limits the computing power of the devices, weakening their resistance to malicious attacks and resulting in frequent malicious attacks on IoT devices. Blockchain is a new computer technology formed by integrating consensus mechanisms [10] and cryptographic algorithms [11]. Its special data structure design enhances data security, prevents malicious tampering, and has characteristics such as decentralization, no trust requirement, and secure data sharing. Therefore, blockchain technology can effectively address the challenges encountered by the IoT in terms of data sharing, privacy protection, and de-trusting between devices [12].

While blockchain technology has the prospect of wide application [13], there are also technical bottlenecks that restrict its development, such as storage capacity issues. Take Bitcoin as an example: as of August 10, 2022, Bitcoin has generated a total of 748,801 blocks, the entire blockchain size is 420.46 GB, and it is growing at a rate of about 58 GB per year [14]. Nodes that want to verify transactions and compete for blocks must download and store a complete copy of the blockchain data, which places high demands on the node's local storage performance. Over time, the node needs to continuously invest storage resources to meet this demand. For IoT applications, hundreds of end devices generate a large amount of data in a short period of time, which is stored and analyzed. If blockchain is applied in the IoT scenario, it is difficult for IoT devices to meet the storage requirements of blockchain due to limited resources for IoT device configuration, and it takes an incredible length of time to put in new IoT devices for data synchronization as the amount of data increases. The Simple Payment Protocol (SPV) [15], which divides nodes into full and lightweight nodes, is a popular lightweight storage method in blockchain. The full node stores a full copy of the block data, is responsible for broadcasting and verifying transactions, and calculates a random number that satisfies the current difficulty to generate new block data. Lightweight nodes store only block headers due to a lack of storage space. Lightweight nodes are all dependent on the full node to perform payment validation, which increases the bandwidth pressure on the full node, can lead to gradual data centralization, and has different degrees of impact on the related application environment [16]. The blockchain network's nodes are distributed all over the world with no central organizational constraints, necessitating incentive mechanisms to encourage nodes to collaborate in order to keep the blockchain system secure. The incentive mechanism is the process of maximizing participants' commitment to the system through specific methods and management. In Bitcoin, incentive mechanisms in the form of virtual coins as rewards encourage nodes to participate actively in the block competition.

In summary, the storage problem in the combination of IoT and blockchain is still a pressing issue, which makes it necessary to study and find ways to reduce the storage pressure on device nodes. Current research on blockchain mainly focuses on consensus mechanisms [17,18], privacy protection [19], and relatively little research on storage optimization of blockchain. Franca [20] proposes a mini-blockchain model for Bitcoin, where nodes store block headers and the latest blocks, and balance information is secured using an account tree, which greatly reduces the storage pressure. However, the model can lead to the loss of transaction data and thus untraceability. Jia et al. [21] proposed a storage capacity scalability model, which determines the number of its storage copies based on block security. Nodes are divided according to their functions, and each node is no longer required to store the complete blockchain information. However, this model uses two auxiliary chains to ensure data security, which increases the complexity of the system. Zheng et al. [22] [The author names in the text citation has been changed to match with the reference list. Kindly verify.] used the Interplanetary File System (IPFS) to reduce the storage pressure on nodes by storing transaction data in IPFS and storing the hashes of the returned transactions in blocks. Compared to the transaction data, its hash value takes up less storage and can effectively reduce the storage pressure on the node, but it increases the complexity of the node to perform transaction verification. Chou et al. [23] divided the block data into hot and cold data based on the number of times the block data was accessed, the cold data was stored in the IPFS system and the hot data was stored in the local disk to reduce the storage pressure on the nodes, but this increased the complexity of verifying the cold data. Dai et al. [24] proposed a blockchain low storage architecture that uses network coding and distributed storage to reduce the storage pressure on nodes, but the scheme only gives a theoretical analysis and no specific implementation plan. Xu et al. [25] proposed an Efficient Public Blockchain Client (EPBC) model, where nodes store blockchain abstracts to verify the correctness of data due to the lack of storage capacity of sensors and mobile devices such as smartphones, but the scheme failed to address the problem of data centricity. Xu et al. [26] proposed the concept of using consensus units, where multiple nodes coordinate and cooperate to store one copy of blockchain data, as a way to reduce the storage space consumption of nodes.

A large number of scholars have carried out a series of works to make IoT technologies better serve people’s lives. Qi et al. [27] proposed an association graph-based recommendation method for Web APIs to be applied to the development of IoT mobile APPs from the perspective of personalization and compatibility. To enable users to find the services they need better and faster, Wang et al. [28] proposed a hybrid collaborative filtering recommendation method to predict the preferences of cold-start users, while some scholars have also conducted research on users’ interest points [29,30]. Yang et al. [31] proposed an anomaly detection method in data streams for the security of IoT devices. Yan et al. [32] proposed a better quality of service prediction method to optimize network service recommendation in mobile edge computing environment [33].

The cloud’s large storage capacity and computing power help users eliminate the stress of storing and maintaining data [34], offload local computing tasks, and share data [35]. Furthermore, the cloud’s convenience and cost effectiveness, which can provide services at a lower cost than midsize data centers, are attracting an increasing number of academics to use the cloud in IoT and blockchain. Cao et al. [36] proposes a multi-cloud framework to address the limitations of medical IoT devices in terms of data access, storage, scalability, and computation. Kim et al. [37] proposes an efficient resource management scheme that uses cloud infrastructure to efficiently manage IoT resources to meet high availability, scalability, and processing volume requirements. Proxy mining and cloud mining methods are used in [38] to cope with problems such as the lack of computational and network resources of device nodes for mining in the IoT, where blockchain is deployed. The block data is stored on a cloud server to alleviate the burden of storing and maintaining data in IoT devices [39]. However, the storage resources consumed by this scheme are not reduced, and the pressure to store the complete ledger of the blockchain is passed to the corresponding servers. In addition, IoT devices generate data at a very high rate, which will put tremendous storage pressure on cloud servers and can even overload them.

Although the current research solutions can alleviate the storage problem of blockchain to some extent, However, none of the schemes consider the issue of whether peers are willing to consume storage space to store data unrelated to themselves. This paper proposes a lightweight storage framework combining IoT and blockchain from the perspective of data autonomy. For each peer, it only needs to store data about the transactions in which they are involved. In addition, the peers can leverage the super computing power of cloud servers to assist in transaction validation. The main contributions made in this paper are as follows:

(1) A lightweight storage framework for blockchain-enabled IoT is proposed to reduce the storage consumption of IoT devices while still allowing transaction verification.
(2) By analyzing the availability and security of data, data redundancy mechanisms are proposed to prevent data loss.
(3) An incentive mechanism is designed based on the reliability of peers to encourage redundant storage of data by peers with spare storage capacity to maintain the stability of the blockchain system.
(4) The superiority of the proposed scheme in terms of performance is demonstrated by an experimental comparison with the traditional blockchain model.

2 Preliminary Knowledge

In this section, the theoretical knowledge related to blockchain is introduced. In addition, the relevant cryptographic knowledge used in the proposed model, such as the strong Rivest-Shamir-Adleman (RSA) assumption, Bezout theorem, accumulator, and RSA accumulator, is also described.

2.1 Blockchain

Blockchain is a shared database technology that uses a distributed architecture and is maintained by multiple parties. The technology is based on a P2P network and requires each node to hold a copy of the data in the ledger to ensure data consistency. The data is stored in blocks and forms a chained structure in chronological order, with the latter block storing the hash value of the previous block. A consensus mechanism is used to select the bookkeeper to generate the block data to update the distributed ledger and achieve distributed consistency. If the transaction data in a block is tampered with, it will lead to a change in the Merkle root and the entire block hash. Therefore, if an attacker wants to tamper with the transactions in a block, he needs to forge a block with the exact same hash value as the previous block, but this process is difficult to implement. In blockchain, data is not written into the distributed ledger immediately after it is generated, but only after it is judged by a series of rules. To introduce the blockchain workflow, let's take Bitcoin as an example. If a value transfer occurs between participants A and B, then the transaction data is created, and the resulting transaction data is first broadcast to all nodes of the network through the P2P network. At first, the nodes of parties that have no transaction receive the transaction data and verify the legitimacy of the transaction. If the transaction is legitimate, it is deposited into the transaction pool. Next, the nodes pack the transactions that are in the transaction pool and compete for prior bookkeeping rights through the consensus algorithm, and the nodes that have obtained prior bookkeeping rights broadcast the packaged new blocks to the entire network. Then, nodes in the network verify the received block data. Finally, if more than half of the nodes confirm that the block is correct, the block is appended to the existing blockchain.

The transfer of value in the blockchain is done through Unspent Transaction Output (UTXO) changes. Unlike account balances, UTXO is non-divisible and is the basic unit for participating in transactions. UTXO cannot be split, but the specified transaction can be completed by adjusting the input and output. As shown in Fig. 1, each transaction contains a transaction input and a transaction output. The transaction input is the UTXO to be consumed, which is derived from the output of the previous transaction, and the transaction output is the newly generated UTXO. The value of the transaction input is equal to the value of the transaction output. If UTXO is less than the target value, multiple UTXOs can be added as input; if UTXO is greater than the target value, you can add your own address as the zero output to complete the transaction. The same transaction output cannot be used as transaction input for two transactions at the same time, as this would lead to a double-spending problem.

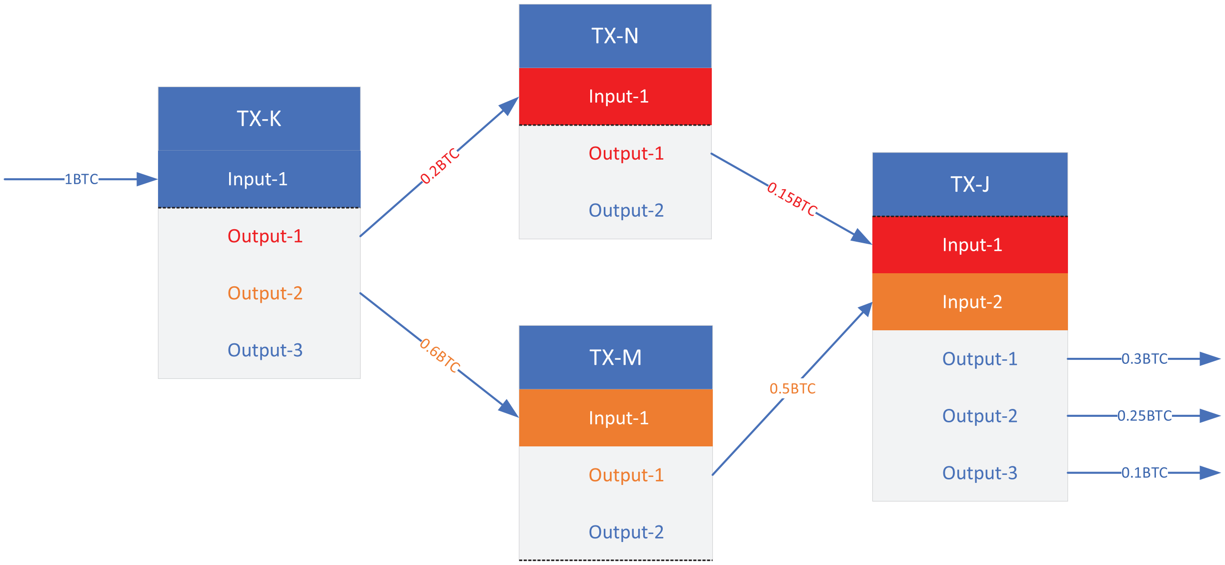


Figure 1: Value transfer process in blockchain

2.2 *Strong RSA Assumptions*

Given an RSA modulo $N = pq$ and a random number $c \in Z_N^*$, Compute $a, b \in Z_N^*$ such that the condition $a^b = cmodN$ (where $b \geq 2$) is satisfied, called the strong RSA problem.

Strong RSA assumption: The strong RSA problem is hard to solve without knowing the factorization of N.

2.3 *Bezout Theorem*

If a, b are integers and $\gcd(a, b) = d$, then for any integers x and y, $ax + by$ must be a multiple of d. In particular, there must exist integers x and y so that $ax + by = d$ holds.

An important inference of this is: the sufficient necessary condition for the coprime of a and b is that there exist certain integers a and b so that $ax + by = 1$ holds.

2.4 *Accumulator*

The accumulator was proposed in 1994, it is mainly used for timestamp and verification of membership. The accumulator is a one-way cryptographic protocol developed through using hash functions, which has a quasi-switching attribute. For all the $x \in X$ as well as $y_1, y_2 \in Y$, the one-way hash function: $h : X \times Y \to X$, satisfies the quasi-switching attribute: $h(h(x, y_1), y_2) = h(h(x, y_2), y_1)$. The accumulator allows to add up the element of the finite set: $X = \{x_1, \ldots, x_n\}$, into a constant value acc_X. With $g \in G$ used as a basic number, the original definition of the accumulator is:

$$acc_X = h(h(h(\ldots h(h(h(g, x_1), x_2), x_3), \ldots, x_{n-2}), x_{n-1}), x_n) \quad (1)$$

Through calculating the demonstration of every element in this set and verifying $h(wit_{x_i}, x_i) = acc_X$, it can effectively demonstrate the membership of the element x_i. Due to the accumulator satisfies the feature of quasi-switching attribute, the cumulative value acc_X not depends on the order of accumulative elements. And due to the collision-free nature of one-way hash function, any element $y \notin X$ that has not been accumulated is infeasible to find a member witness in terms of computation.

2.5 RSA Accumulator

The RSA accumulator is an extension of the original accumulator that allows elements of a set to be added or removed dynamically. The result of accumulation for the set $X = \{x_1, \ldots, x_n\}$ can be expressed as:

$$acc_X = g^{\prod\limits_{i=1}^{n} x_i} \bmod N \tag{2}$$

N is the modulus of the accumulator, which is obtained by multiplying two p and q selected. g is a number randomly sampled from the quadratic cyclic group of modulo N and $\{x_1, \ldots, x_n\}$ is restricted to primes due to the exponential multiplicative relationship between the cumulative values in the computation. The x_i uses wit_{x_i} to prove its existence in the set X where $wit_{x_i} = acc_X^{x_i^{-1} \bmod \phi(N)} \bmod N$. The addition of elements in the accumulator is relatively simple and can be updated quickly without knowing p and q, $acc'_X = acc_X^{x'} \bmod N$, and membership proofs are similarly updated with just $wit'_{x_i} = wit_{x_i}^{x'} \bmod N$. Bezout theorem can be used to update the membership proof when removing a set member. For example, if member x_j is deleted, the membership proof of x_i can be expressed as $wit'_{x_i} = wit_{x_i}^{b} (acc'_X)^a \bmod N$. Where a and b satisfy $ax_i + bx_j = 1$ and acc'_X is the updated cumulative value. It is obvious that the membership proof of x_j is equal to acc'_X.

3 System Description and Model

Traditional IoT devices are cost-constrained, resulting in scarce storage resources and weak computing power. However, the requirement of blockchain technology for high storage and high computing power among peers in the system has limited the development of blockchain-enabled IoT. In view of making blockchain technology applicable to IoT applications, this section describes the proposed lightweight storage framework in terms of storage strategy and redundancy mechanisms, an incentive model, and a transaction verification model, respectively.

3.1 System Model

The system model is designed with three objectives: 1) the distributed storage nature of the blockchain is not changed. 2) There is no assumption of trust between devices in the system. 3) The peers reduce storage consumption while possessing the function of transaction verification. The system model is shown in Fig. 2. Each IoT device corresponds to a peer in the blockchain network, and after the IoT device generates a transaction, the peer uses the supercomputer power of the cloud server to verify the transaction, which is verified by depositing it into the transaction pool. The peers then compete for priority bookkeeping rights to generate the block data. After the block data is generated, the peers select the transaction data they participate in and some of the interested transaction data for storage, where the degree of interest of the peers in the transactions is influenced by the incentive model and their own storage capacity. The greater the number of transactions in which an IoT device is involved, the more data it stores. However, new devices are not involved in transactions and spend much less time synchronizing data when they join the system. Peers store the transaction data of their own participation instead of storing the complete block data, which makes the data storage capacity of the whole blockchain system less correlated with the number of devices and effectively improves the storage scalability of the system.

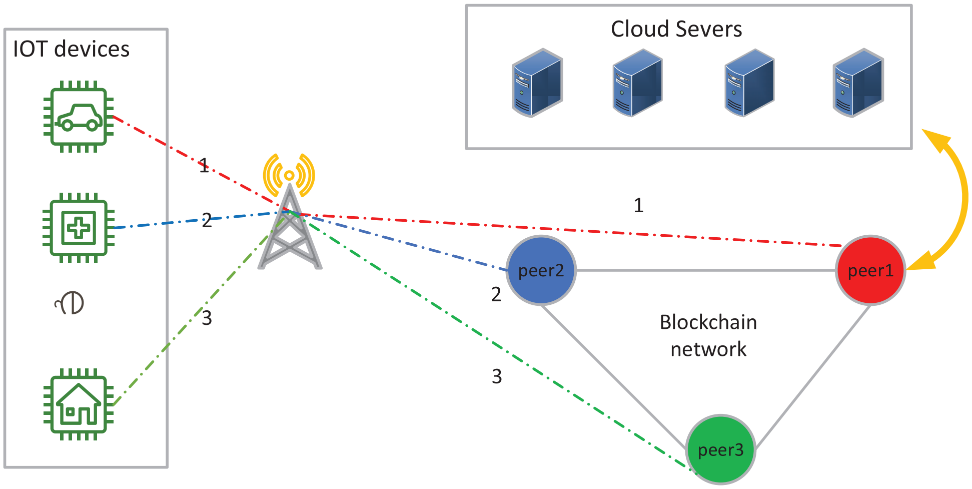


Figure 2: System model

Assuming that peers only store the transaction data of their own participation, there may be a situation where all the participants of a transaction withdraw from the blockchain system, and the transaction data is lost. Peers thus retain transaction data from other nodes, maintaining both the integrity of the blockchain system and the non-loss of transaction data. In addition, peers need to store block headers to prevent block data from being tampered with.

The selective retention of transaction data by the peers effectively maintains the distributed data storage characteristics of the blockchain system and reduces storage consumption, but also brings about various problems, such as transaction validation. Hence, the system needs to consider the following issues: 1) What storage strategy should be set by the model to achieve data redundancy and maintain the benign ecology of the blockchain system. 2) how to motivate peers to pay for resources to store transaction data that is not related to themselves. 3) how to verify transactions with the help of cloud servers when peers selectively keep transaction data, which will result in not having a complete UTXO set to verify transactions. Therefore, this paper will continue to elaborate on the storage strategy, transaction validation model, and incentive mechanism to solve the above three problems.

3.2 Storage Strategy and Redundancy Mechanism

Peers selectively store transaction data, and all peers in the blockchain network implement distributed storage of blockchain data. The peers store transaction data as shown in Fig. 3. Compared with storing all transaction data, peers only store tx-4 and the Merkle branch of tx-4, and the data of tx-1, tx-2, and tx-3 are strategically dropped from storage. The Merkle branch of the peer storage tx-4 is used to prove that the tx-4 data has not been tampered with in transaction validation due to the Merkle tree's ability to quickly summarize and verify the integrity of block data. While peers selectively store transaction data to reduce storage consumption, they can send data requests to other peers, such as jigsaw puzzles, to restore block data.

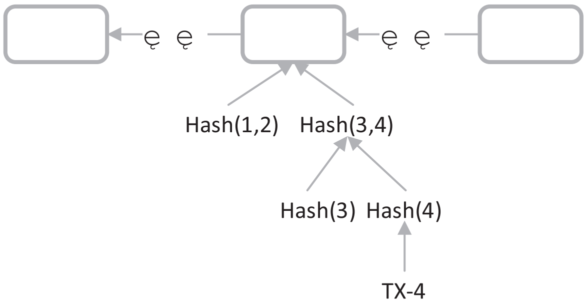


Figure 3: Transaction data storage strategy

Peers maintain account balances by storing their own transaction data. If the transaction data is only stored by the participating peers, this will result in too few copies of the transaction data, which can easily lead to transaction data loss. Therefore, the peers need to store the transaction data redundantly to avoid the loss of transaction data in the blockchain system. In the P2P network, the redundancy mechanism can be set according to the reliability of the peers, and the reliability P_r of the peers is shown in Eq. (3).

$$P_r = \frac{1}{T_w} \int_0^t (1 - P[T_w < u])\, du \tag{3}$$

T_w is the single peer failure time and t is the fixed time for other peers to send requests to that peer. The data is lost if all peers that are simultaneously storing the same data collapse. the probability P_s of data loss is as shown in Eq. (4).

$$P_s = \prod_{i=1}^{N} \left(1 - \frac{1}{T_w^{\,i}} \int_0^t (1 - P[T_w^{\,i} < u])\, du \right) \tag{4}$$

N is the number of peers storing the same data, $T_w^{\,i}$ is the ith peer's average failure time among the N peers. Combining Eqs. (3) and (4), it can be concluded that the availability P_a of data in a P2P network is:

$$P_a = 1 - \prod_{i=1}^{N} \left(1 - \frac{1}{T_w^{\,i}} \int_0^t (1 - P[T_w^{\,i} < u])\, du \right) \tag{5}$$

From Eq. (5), it is clear that the more copies of transaction data and the higher reliability of the peers will lead to greater data availability. The transaction data being stored redundantly effectively ensures the availability of the data, but peers are often reluctant to pay for extra storage resources to record transaction data that are not relevant to them. Therefore, the question of how to incentivize peers to store transaction data redundantly has become an urgent problem.

3.3 Incentive Mechanism

Rather than store all the transaction data without a strategy, the peers store the transaction data selectively according to the incentive mechanism. As peers store less data, new challenges are posed for peers to validate transactional data. In the Bitcoin system, by storing the full block data and thus obtaining the UTXO set, peers can verify the legitimacy of transactions. In contrast, in the research for this paper, peers store block data selectively, so peers cannot obtain all UTXO data locally for data validation.

According to Eq. (5), The availability of data P_a is mainly determined by the reliability P_r of the peers and the number of copies N of data. We can use a monitoring tool such as Ethstats [40] in Ether to monitor the peers in the system in real time to get the reliability of the peers. Knowing the availability P_a of the data and the reliability P_r of the peers, the number of copies N of the data can be obtained according to Eq. (5). Under high availability of data, the number of copies of data can be expressed as N_a.

To attract highly reliable peers to actively participate in storing redundant data, the incentives are designed based on the reliability of the peers and the number of copies of the data. Peers with high reliability who are the first to store transaction data are rewarded more. Since peers need to store their own participating transaction data to maintain their account balances, transaction participants do not participate in the revenue distribution of these transaction data. $\omega_{\max}$ is defined as the highest value of the revenue available to the peer, which represents the revenue obtained by the absolutely reliable peer being the first to store the transaction data. $\omega_{\min}$ is defined as the cost of storage resources consumed by the peer to store the data. The revenue received C_i by the peers for storing transaction data is shown in Eq. (6).

$$C_i = \begin{cases} P_r\omega_{\max}\left(1 - \frac{N_m - N_n}{N_a - N_n}\right) + \omega_{\min} & \text{if } N_m - N_n \le N_a - N_n \\ \omega_{\min} & \text{if } N_m - N_n > N_a - N_n \end{cases} \tag{6}$$

where N_m is the number of copies of the transaction data that are stored, and N_n is the number of copies of the transaction participant. From Eq. (6), the revenues obtained by the peers are mainly influenced by the reliability of the peers and the amount of existing data. Therefore, in order to obtain high revenue, the peers must maintain high reliability while also actively engaging in redundant data storage. And when the number of transaction storage copies has reached N_a, peers with abundant storage resources can still use the free resources to store transaction data to get some bonus. Under the incentive mechanism, peers will strive to maintain their high reliability and actively store transaction data in order to obtain high revenue. The availability of data is effectively guaranteed.

3.4 Transaction Validation Model

The peers store the transaction data selectively based on the incentive mechanism, not storing all the transaction data without policy. The reduction in storage poses new challenges for peers to verify transaction data. In the Bitcoin system, peers verify the legitimacy of transactions by storing the full block data and thus obtaining the UTXO set. In this research, the peer cannot get all UTXO data locally for data validation because the peer stores block data selectively.

To enable peers to store small amounts of data while still participating in transaction verification, this research draws on the work of Boneh et al. [41] to design a transaction verification model using RSA accumulators. The RSA accumulator can accumulate a collection of elements into a constant-size digest to verify element signatures and can dynamically add and remove elements from the collection. In blockchain systems, the UTXO set is constantly changing as transaction data is generated. Hence, the combination of UTXO and RSA accumulator can solve the problem of peers storing small amounts of transaction data that cannot be verified locally using traditional methods for transaction validation. However, the computational performance of IoT devices is often weak and cannot be adapted to the RSA accumulator application scene. IoT devices can verify transactions with the powerful computing power provided by the cloud server, and the transaction verification process is shown in Fig. 4.

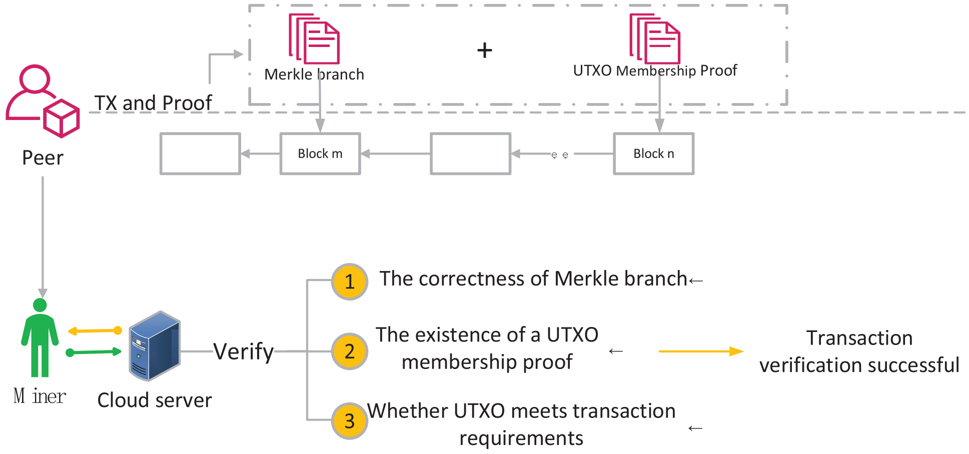
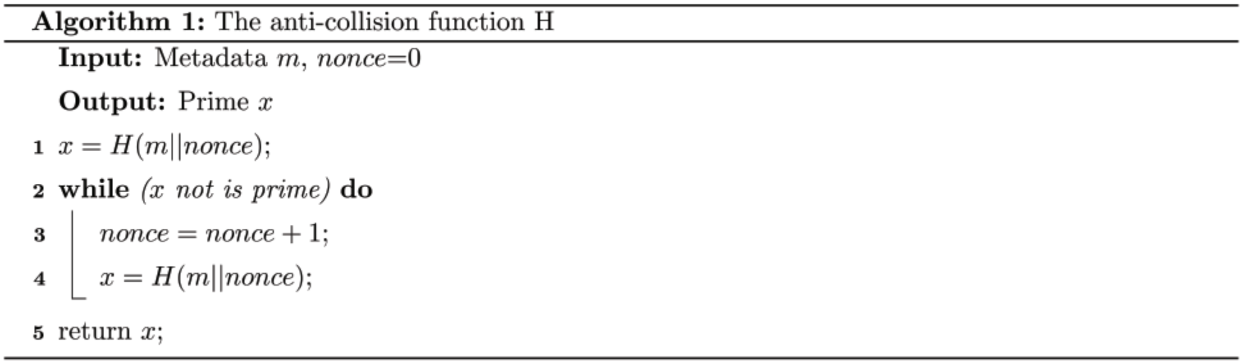

TX and Proof
Merkle branch
+
UTXO Membership Proof
Peer
Block m
Block n
Miner
Cloud server
Verify
1 The correctness of Merkle branch←
2 The existence of a UTXO membership proof ←
3 Whether UTXO meets transaction requirements ←
Transaction verification successful

Figure 4: Transaction verification process

In transaction validation, peers broadcast transaction data and proof data. The proof data contains the UTXO membership proof of the transaction input and the Merkle branch. If the input UTXO for the current transaction originates from tx-4, the Merkle branch containing tx-4 is sent. The peers competing out of the block receive the broadcasted data and use the cloud server to verify the transaction's legitimacy. The cloud server first verifies the correctness of the Merkle branch. The verification process is similar to the SPV. Specifically, the Merkle branch from the leaf to the root is verified layer by layer and then compared with the root of the tree in the block header. Second, the existence of the UTXO is verified by UTXO membership proof to prevent the UTXO from being double-spent. Finally, the input UTXO is verified to meet the transaction requirements, and if all the verifications pass, the transaction is legitimate. A coinbase transaction is included in the block data, which is used to reward miners for their contributions. Unlike regular transactions, coinbase transactions have no input and do not consume UTXO. The coinbase transaction has an output that pays to the address of this miner. During transaction validation, the miner does not validate the coinbase transaction, but adds the coinbase transaction directly to the block data. In the transaction verification process, no assumption of trust between peers is achieved because the transaction proponent can provide its own proof without relying on other peers.

The most important step in transaction validation is to verify the existence of UTXO with the help of RSA accumulator. In the RSA accumulator, the virtual quadratic order group [41] is used to avoid the initialization of setting two large prime numbers p and q. This is due to the fact that the blockchain environment is untrusted and data forgery is possible if there exists a peer that knows p and q. In addition, the elements in the RSA accumulator have prime limits and the anti-collision function $H(0,1)^* \rightarrow Z_N^*$ is designed to perform prime conversions. $\{UTXO_1, UTXO_2 \ldots UTXO_n\}$ is transformed into $\{x_1, x_2 \ldots x_n\}$ by the anti-collision function H. According to Eq. (2), all the UTXOs are added to the RSA accumulator to obtain the accumulation value acc_X of the UTXO set. The membership proof of $UTXO_i$ can be expressed as $wit_{x_i} = acc_X^{x_i^{-1} \bmod f(N)} \bmod N$. The peers involved in mining only need to calculate $(wit_{x_i})^{x_i} \bmod N = acc_X$ to determine whether $UTXO_i$ is spent or not.

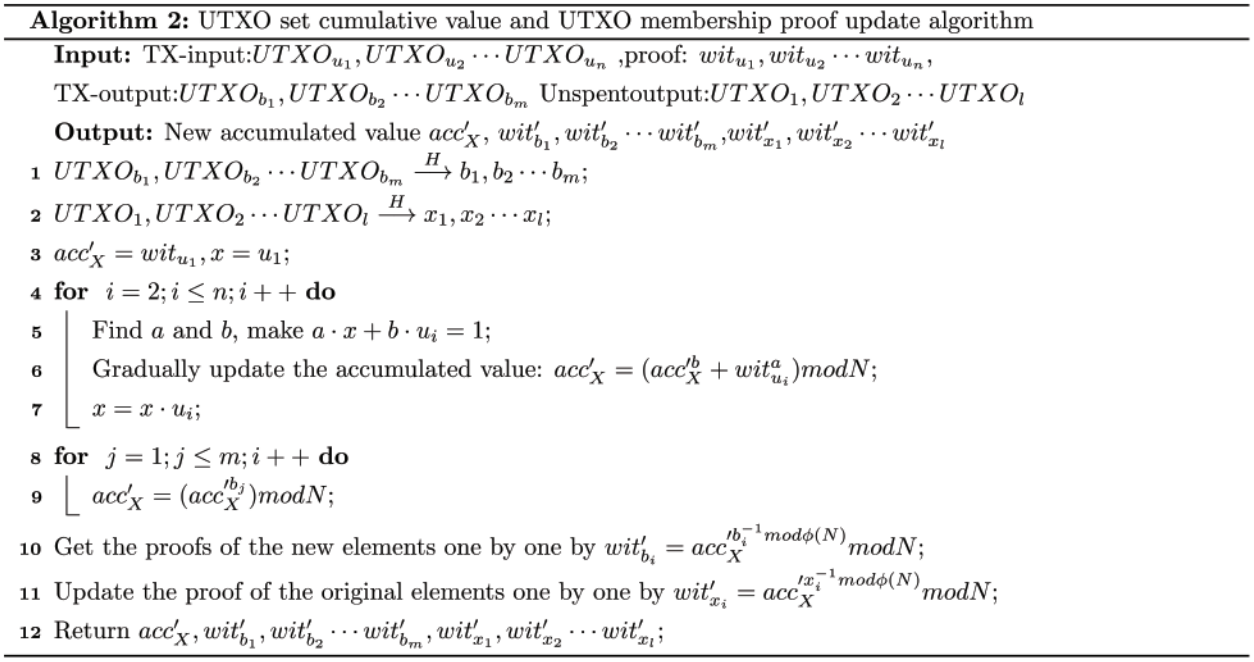


```
Algorithm 1: The anti-collision function H
Input: Metadata $m$, $nonce$=0
Output: Prime $x$
$x = H(m||nonce)$;
while ($x$ not is prime) do
    $nonce = nonce + 1$;
    $x = H(m||nonce)$;
return $x$;
```

The transaction initiator broadcasts the transaction data, which is then validated by the peers and deposited into the transaction pool, and finally packaged into block data, which is accompanied by old UTXOs being used and new UTXOs being created. The members of the accumulator are constantly changing, so to verify that UTXO is spent, the accumulation value acc_X and the UTXO membership proof should be updated in time after the block data is generated. In the RSA accumulator, when adding a member variable x_i, the change of the accumulation value is done by $acc_X^{'} = acc_X^{x_i}\mathrm{mod}N$, and the update of the member proof is also done by $wit_{x_j}^{'} = wit_{x_j}^{x_i}\,\mathrm{mod}N$. If member x_i is to be deleted, the cumulative value of the accumulator is the proof of x_i. Therefore, the update of the cumulative value after deleting a member can be done by aggregating the proofs of the members to be deleted. Taking aggregation x_i and x_j as an example, the proof of aggregation of x_i and x_j can be calculated as $Wit_{ij} = (wit_{x_i}{}^{b} + wit_{x_j}{}^{a})\mathrm{mod}N$, where a and b satisfy $ax_i + bx_j = 1$, which is due to the fact that x_i and x_j are mutually prime and the values of a and b can be found by Bezout theorem. The verification of the proof of aggregation of x_i and x_j is shown in Eq. (7). Since the members are all prime, the new cumulative value acc'_X is obtained by simply aggregating the proofs of the members to be deleted one by one. The update of the accumulator requires adding or removing members one by one, so there is no situation where the balance of the peers remains unchanged and the composition of the balance changes.

$$acc_X{}^{'x_i x_j} = Wit_{ij}^{x_i x_j} = (wit_{x_i}{}^{b} + wit_{x_j}{}^{a})^{x_i x_j} = wit_{x_i}{}^{bx_i} + wit_{x_j}{}^{ax_j} = acc_X{}^{ax_i + bx_j} = acc_X \tag{7}$$

In the cumulative value and UTXO membership proof update algorithm, the latest cumulative value of the UTXO set is obtained by aggregating the member proofs of the spent UTXO and adding the newly generated UTXO to the accumulator. UTXO membership proofs are updated with the latest cumulative values. This application scenario, where the UTXO set is constantly changing in the blockchain environment, is satisfied by updating the cumulative value and membership proof. The transaction verification model can effectively determine whether the UTXO has been spent and prevent the double-spending problem. The peer implements not owning all of the block data, but still being able to perform transaction validation.



Algorithm 2: UTXO set cumulative value and UTXO membership proof update algorithm

```
Input: TX-input:$UTXO_{u_1}, UTXO_{u_2} \cdots UTXO_{u_n}$ ,proof: $wit_{u_1}, wit_{u_2} \cdots wit_{u_n}$,
TX-output:$UTXO_{b_1}, UTXO_{b_2} \cdots UTXO_{b_m}$ Unspentoutput:$UTXO_1, UTXO_2 \cdots UTXO_l$
Output: New accumulated value $acc'_X$, $wit'_{b_1}, wit'_{b_2} \cdots wit'_{b_m}, wit'_{x_1}, wit'_{x_2} \cdots wit'_{x_l}$
$UTXO_{b_1}, UTXO_{b_2} \cdots UTXO_{b_m} \xrightarrow{H} b_1, b_2 \cdots b_m$;
$UTXO_1, UTXO_2 \cdots UTXO_l \xrightarrow{H} x_1, x_2 \cdots x_l$;
$acc'_X = wit_{u_1}, x = u_1$;
for $i = 2; i \leq n; i++$ do
    Find $a$ and $b$, make $a \cdot x + b \cdot u_i = 1$;
    Gradually update the accumulated value: $acc'_X = (acc'^b_X + wit^a_{u_i}) mod N$;
    $x = x \cdot u_i$;
for $j = 1; j \leq m; i++$ do
    $acc'_X = (acc'^{b_j}_X) mod N$;
Get the proofs of the new elements one by one by $wit'_{b_i} = acc'^{b_i^{-1} mod \phi(N)}_X mod N$;
Update the proof of the original elements one by one by $wit'_{x_i} = acc'^{x_i^{-1} mod \phi(N)}_X mod N$;
Return $acc'_X, wit'_{b_1}, wit'_{b_2} \cdots wit'_{b_m}, wit'_{x_1}, wit'_{x_2} \cdots wit'_{x_l}$;
```

4 Experiment and Evaluation

The experiment was run on a PC with an 8-core Intel i5-11320H 3.20 GHz processor and 16 GB memory. The related performance of the model was analyzed based on the RSA accumulator, blockchain source code, and cloud server. The RSA accumulator is set using parameters recognized as secure in the field of cryptography, with the modulus set at 3072 bits and the prime at 128 bits. The experiment tests and analyzes the data availability, incentive mechanism, storage optimization effect, and time consumption. The data availability experiment mainly tests the redundant storage of data by peers with different reliability and the change in data availability. The incentive mechanism tests the profit from storing data among peers with different reliability. Storage optimization primarily verifies peer storage reduction under different transaction numbers of block data. The time consumption experiment mainly tests the time consumption of the cumulative value update and transaction verification of the RSA accumulator after the new block is generated.

4.1 Data Availability

To test the effect of redundancy mechanism to enhance data availability, we virtualized 600 peers using Docker technology and randomly marked 150 peers as a group of observation peers, for a total of 4 groups of observation peers. The reliability setting of each group of observation peers is different, which is the random number in different intervals, and the reliability setting interval includes [0∼1], [0.1∼1], [0.2∼1] and [0.3∼ 1]. In order to better demonstrate the effect of redundancy mechanism to enhance data availability, the reliability of each group of observed peers is arranged in ascending order. The peer points store the transaction data one after another in the order of reliability to obtain the relationship between data redundancy and data availability, and the results are shown in Fig. 5.

The experimental results in Fig. 5 show that the availability of transaction data improves as the number of data stores increases, while the initial availability of transaction data also increases as the original value of the peers increases. On the other hand, the higher the initial value of the peer reliability interval, the higher the slope of the availability variation curve before reaching the stable region, which indicates that the higher the peer reliability, the fewer redundant data copies are needed to guarantee the availability of transaction data. The subsequent experiments use the worst case, i.e., the case with the smallest initial value of the pair-end peer reliability, i.e., the interval [0, 1], as the basis for validation. At this point, the data availability tends to 100% when the number of peers storing data reaches 50.

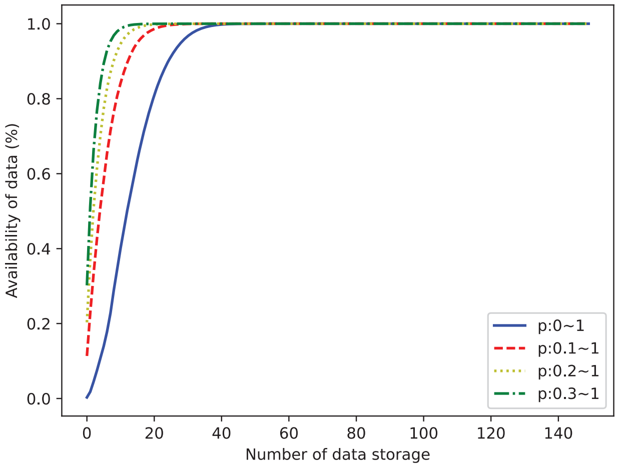


Figure 5: Data availability

4.2 Incentive Mechanism

The purpose of the incentive mechanism is to encourage peers to actively store transactions that is not relevant to them in order to improve data availability. To motivate peers and attract highly reliable peers to store transaction data, the mechanism regulates the revenue based on the number of transaction data stored and the reliability of the peers. In the data availability experiment, the data availability tends to be 100% when the number of stored copies reaches 50, so N_a is set to 50 and the data is not considered to be stored by the transaction party. In the experiment, $\omega_{\max}$ is set to 20, $\omega_{\min}$ is set to 5, and 55 peers are selected to store the transaction data. The reliability of the peers is a random number between 0 and 1. The peers store the data in random order, and the revenue is as shown in Fig. 6.

The benefits of peers are mainly influenced by reliability and the number of data stores. The trend of peer's revenue change is the same as the trend of peer's reliability change, but as the number of data stores increases, the impact of peer's reliability change on revenue also diminishes. Therefore, there are two scenarios: peers with high reliability who benefit less than peers who store data initially, and peers who store data first but benefit less than peers with high reliability. Moreover, when the number of data copies reaches the data availability requirement, the peers obtain a benefit equal to the value of the consumed storage resources. Driven by incentives to maximize benefits, peers need to maintain a high level of reliability while actively participating in storing data.

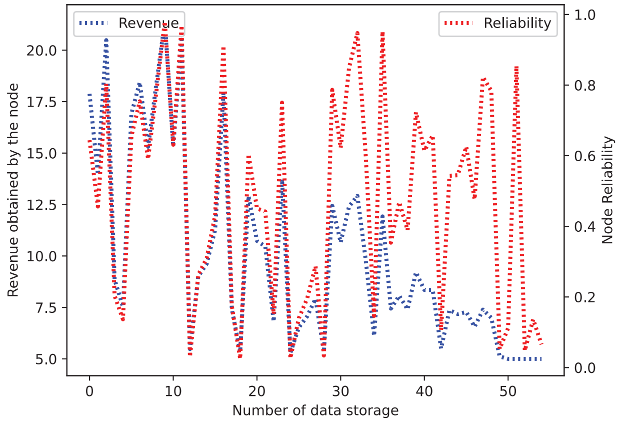


Figure 6: Relationship between peer's revenue and reliability

4.3 Storage Consumption

In this scenario, the storage consumption of the peers is mainly affected by the storage consumption of transaction data and the storage consumption of UTXO membership proofs, so the tests evaluate the storage consumption of the peers from these two aspects separately. The transaction data stored by the peers includes transaction data from its own participation and other transaction data stored redundantly. As the other transaction data stored by the peers is for the purpose of obtaining revenues, redundant storage of data is not considered in the storage consumption test. When peers store transaction data, they also need to store block headers and the corresponding Merkle branches. The experiments put the peer storage consumption to the test by generating blocks with varying amounts of transaction data. The number of transactions contained in the blocks is 8, 32, 64, 128, and 256, respectively. It is known from the storage strategy that peers tend to store only the transaction data they are involved in, so when a peer receives the broadcasted block data, the block data needs to be clipped. For a better evaluation of the storage optimization of the scheme, the peers are involved in only one transaction data within the block during the block generation process, i.e., the peers only need to store one transaction data, and the experimental results are shown in Fig. 7.

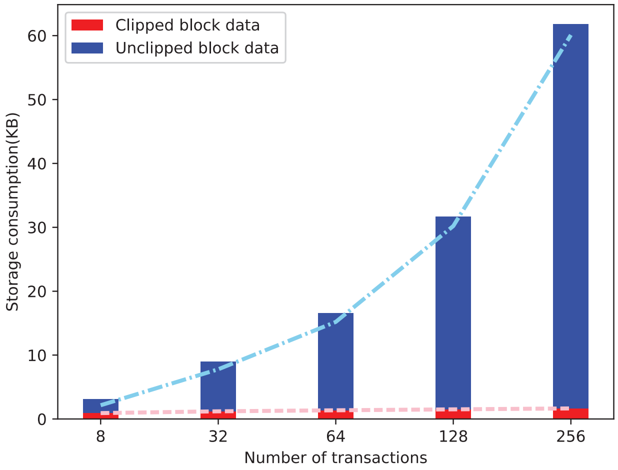


Figure 7: Storage consumption comparison

Experimental results show that the peers effectively reduce storage consumption by trimming the block data. As the number of transactions gradually increases from 8, 32, 64, 128, 256, the reduction in storage consumption of the block data stored on the peer side can reach 57.21%, 84.47%, 91.05%, 95.00%, 97.24% respectively. In transaction validation, the peers involved in mining do not store the complete blockchain data, so the peers need to provide UTXO membership proofs to verify that the UTXO has not been spent. The modulus of the RSA accumulator in the scheme is set to 3072 bits, which makes the size of each UTXO membership proof constant at 384 bytes. Peers maintain account balances by holding one or more UTXO membership proofs, so the storage consumption of UTXO membership proofs is an integer multiple of 384 bytes. In summary, compared with the traditional blockchain model, the proposed framework can better reduce the storage consumption of peers and increase the storage scalability of the system.

4.4 Time Consumption

In the transaction validation model, the peers perform transaction validation with the help of cloud servers. The peer verifies the existence of the transaction input and whether the transaction requirements are satisfied based on the source Merkle branch of the transaction input UTXO and the UTXO membership proof. After the block data is generated, the latest UTXO set accumulation values and UTXO membership proofs are updated, relying on the superb computing power of the cloud servers. The time consumption of transaction validation is mainly composed of the UTXO set accumulation values update time and the transaction input validation time, so the experiment uses a cloud server to test the time consumption of UTXO set accumulation values update and transaction input validation in the proposed framework. The experiments generate block data containing a different number of transactions, where each transaction contains 3 input UTXOs and 3 output UTXOs, so as to test the time consumption of updating UTXO set accumulation values with a different number of transactions, and the experimental results are shown in Fig. 8.

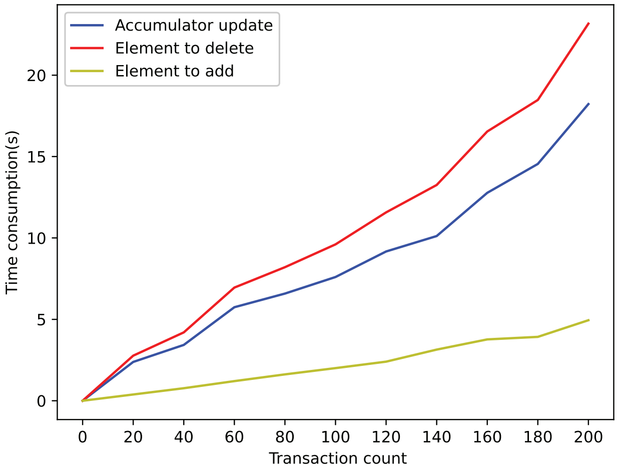


Figure 8: Accumulator update time consumption

Fig. 8 shows that the cumulative value update time of the UTXO set gradually becomes larger as the number of UTXOs involved in the block increases. The deletion operation is the main factor affecting the update time, and the more UTXOs need to be deleted, the longer the UTXO set cumulative value update time of the UTXO set. When the number of transactions in the block is 200, the number of UTXOs involved in the block is 1200, and the throughput of UTXO verification is 51.8TPS at this time, which has a better performance advantage compared to traversing the block data for transaction verification.

5 Conclusion

In this paper, we propose a lightweight storage framework to address the storage challenges faced by blockchain-enabled IoT. Peers only need to store the transaction data of their own participation and verify the transactions with the supercomputing power of cloud servers. In addition, to guarantee the availability and integrity of transaction data, we design incentive mechanisms to motivate peers to store redundant data. Compared with the traditional framework of blockchain-enabled IoT, which uploads block data to the cloud, the proposed framework effectively reduces the storage overhead and communication overhead of the whole blockchain system and greatly reduces the time consumption of data synchronization when new peers join. At the same time, the framework has an advantage in UTXO verification throughput. In the future, the incentive mechanism can be studied further to prevent malicious attacks. In addition, improving the update efficiency of RSA accumulators can be a new research direction.

Acknowledgement: We would also thank the support from the National Natural Science Foundation of China (Nos. 62172182, 62202118, 61962009), the Top Technology Talent Project from Guizhou Education Department (Qian jiao ji [2022]073), The Opening Foundation of Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province (Grant No. ZNKZN2021-07).

Funding Statement: This work is supported by the National Natural Science Foundation of China (Nos. 62172182, 62202118, 61962009), the Top Technology Talent Project from Guizhou Education Department (Qian jiao ji [2022]073), The Opening Foundation of Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province (Grant No. ZNKZN2021-07).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Tong, F. Ye, M. Yan, H. Liu and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[2] F. Wang, G. Li, Y. L. Wang, W. Rafique, M. R. Khosravi *et al.,* "Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city," *ACM Transactions on Internet Technology*, 2022. https://doi.org/10.1145/3511904

[3] X. L. Xu, H. Y. Li, W. J. Xu, Z. J. Liu, L. Yao *et al.,* "Artificial intelligence for edge service optimization in internet of vehicles: A survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[4] J. Dong, W. Wu, Y. Gao, X. Wang and P. Si, "Deep reinforcement learning based worker selection for distributed machine learning enhanced edge intelligence in internet of vehicles," *Intelligent and Converged Networks*, vol. 37, no. 1, pp. 135–151, 2022.

[5] Y. W. Liu, D. J. Li, S. H. Wan, F. Wang, W. C. Dou *et al.,* "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 1, no. 3, pp. 234–242, 2020.

[6] L. Y. Qi, Y. H. Yang, X. K. Zhou, W. Rafique and J. Ma, "Fast anomaly identification based on multi-aspect data streams for intelligent intrusion detection toward secure industry 4.0," *IEEE Transactions on Industrial Informatics*, 2022. https://doi.org/10.1109/TII.2021.3139363

[7] B. Varun, *IoT Trends in 2020: 5 Things You Need to Know*, 2020. [Online]. Available: https://www.iot-forall.com/iot-devices-by-2020

[8] L. Z. Kong, L. Wang, W. W. Gong, C. Yan, Y. C. Duan *et al.,* "LSH-aware multitype health data prediction with privacy preservation in edge environment," *World Wide Web Journal*, vol. 25, no. 5, pp. 1793–1808, 2022.

[9] Y. L. Chen, J. Sun, Y. X. Yang, T. Li, X. X. Niu *et al.,* "PSSPR: A source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.

[10] T. Li, Y. L. Chen, Y. L. Wang, Y. L. Wang, M. H. Zhao *et al.,* "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, 2020. https://doi.org/10.1155/2020/8839047

[11] Y. L. Chen, S. Dong, T. Li, Y. L. Wang and H. Y. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.

[12] A. P. Singh, N. R. Pradhan, A. K. Luhach, S. Agnihotri, N. Z. Jhanjhi *et al.,* "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5779–5789, 2021.

[13] R. Rajmohan, T. A. Kumar, M. Pavithra and S. G. Sandhya, "Blockchain: Next-generation technology for industry 4.0," in *Blockchain Technology*, Boca Raton, FL, USA: CRC Press, 2020.

[14] Blockchain, *The blockchain data of bitcoin* [Online]. Available: https://www.blockchain.com/

[15] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system* 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[16] F. Yuan, S. Chen, K. Liang and L. Xu, "Research on the coordination mechanism of traditional Chinese medicine medical record data standardization and characteristic protection under big data environment," Shandong, CHN, Shandong People's Publishing House, 2021.

[17] T. Li, Z. J. Wang, G. Y. Yang, Y. Cui, Y. L. Chen *et al.,* "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, pp. 3596-3612, 2021.

[18] T. Li, Z. J. Wang, Y. L. Chen, C. M. Li, Y. L. Jia *et al.,* "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10576–10597, 2022. https://doi.org/10.1002/int.22656

[19] Y. Wang, T. Li, M. Liu, C. Li and H. Wang, "STSIIML: Study on token shuffling under incomplete information based on machine learning," *International Journal of Intelligent Systems*, vol. 37, pp. 11078–11100, 2022. https://doi.org/10.1002/int.23033

[20] B. F. Franca. *Privacy and Pruning in the Mini-Blockchain*. 2014. [Online]. Available: http://cryptonite.info/files/Anonymity_account_tree.pdf

[21] D. Jia, J. Xin, Z. Wang, W. Guo and G. Wang, "Storage capacity scalable model for blockchain," *Journal of Frontiers of Computer Science & Technology*, vol. 12, no. 4, pp. 525–535, 2018.

[22] Q. Zheng, L. Yi, C. Ping and X. Dong, "An innovative IPFS-based storage model for blockchain," in *Proc. IEEE/WIC/ACM WI.*, Santiago, Chile, pp. 704–708, 2018.

[23] I. T. Chou, H. H. Su, Y. L. Hsueh and C. W. Hsueh, "BC-store: A scalable design for blockchain storage," in *Proc. IECC*, Singapore, pp. 33–38, 2020.

[24] M. Dai, S. Zhang, H. Wang and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22970–22975, 2018.

[25] L. Xu, L. Chen, Z. Gao, S. Xu and W. Shi, "EPBC: Efficient public blockchain client for lightweight users," in *Proc. SERIAL*, Las Vegas, NV, USA, pp. 1–6, 2017.

[26] Z. Xu, S. Han and L. Chen, "CUB: A consensus unit-based storage scheme for blockchain system," in *Proc. DE*, Piscataway, NJ, USA, pp. 173–184, 2018.

[27] L. Y. Qi, W. M. Lin, X. Y. Zhang, W. H. Dou, X. L. Xu *et al.,* "A correlation graph based approach for personalized and compatible web APIs recommendation in mobile APP development," *IEEE Transactions on Knowledge and Data Engineering*, 2022. https://doi.org/10.1109/TKDE.2022.3168611

[28] F. Wang, H. B. Zhu, G. Srivastava, S. C. Li, M. R. Khosravi *et al.,* "Robust collaborative filtering recommendation with user-item-trust records," *IEEE Transactions on Computational Social Systems*, 2021. https://doi.org/10.1109/TCSS.2021.3064213

[29] Y. W. Liu, Z. L. Song, X. L. Xu, W. Rafique, X. Y. Zhang *et al.,* "Bidirectional GRU networks-based next POI category prediction for healthcare," *International Journal of Intelligent Systems*, vol. 37, no. 7, pp. 4020–4040, 2022.

[30] L. Y. Qi, Y. W. Liu, Y. L. Zhang, X. L. Xu, M. Bilal *et al.,* "Privacy-aware point-of-interest category recommendation in internet of things," *IEEE Internet of Things Journal*, 2022. https://doi.org/10.1109/JIOT.2022.3181136

[31] Y. H. Yang, X. Yang, M. Heidari, G. Srivastava, M. R. Khosravi *et al.,* "ASTREAM: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment," *IEEE Transactions on Network Science and Engineering*, 2022. https://doi.org/10.1109/TNSE.2022.3157730

[32] C. Yan, Y. K. Zhang, W. Y. Zhong, C. Zhang and B. G. Xin, "A truncated SVD-based ARIMA model for multiple QoS prediction in mobile edge computing," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 315–324, 2022.

[33] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181–198, 2020.

[34] A. K. Sandhu, "Big data with cloud computing: Discussions and challenges," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40, 2022.

[35] R. Mendes, T. Oliveira, V. V. Cogo, N. Neves and A. Bessani, "CHARON: A secure cloud-of-clouds system for storing and sharing big data," *IEEE Transactionson Cloud Computing*, vol. 9, no. 4, pp. 1349–1361, 2019.

[36] R. H. Cao, T. Zhuo, C. B. Liu and B. Veeravalli, "A scalable multicloud storage architecture for cloud-supported medical internet of things," *IEEE Internet of Things Journal*, vol. 7, pp. 1641–1654, 2020.

[37] H. W. Kim, J. H. Park and Y. S. Jeong, "Efficient resource management scheme for storage processing in cloud infrastructure with internet of things," *Wireless Personal Communications*, vol. 91, no. 4, pp. 1635–1651, 2016.

[38] C. Qiu, H. Yao, C. Jiang, S. Guo and F. Xu, "Cloud computing assisted blockchain-enabled internet of things," *IEEE Transactions on Cloud Computing*, 2019. https://doi.org/10.1109/tcc.2019.2930259

[39] Y. J. Ren, Y. Leng, J. Qi, P. K. Sharma, J. Wang *et al.,* "Multiple cloud storage mechanism based on blockchain in smart homes," *Future Generation Computer Systems*, vol. 115, pp. 304–313, 2021.

[40] Github. [Online]. Available: https://github.com/swarm-pit/ethstats/

[41] D. Boneh, B. Bünz and B. Fisch, "Batching techniques for accumulators with applications to IOPs and stateless blockchains," in *Proc. IACR*, Santa Barbara, CA, USA, pp. 561–586, 2019.