Hindawi
Security and Communication Networks
Volume 2021, Article ID 5510586, 15 pages
https://doi.org/10.1155/2021/5510586

Research Article

ECLB: Edge-Computing-Based Lightweight Blockchain Framework for Mobile Systems

Qingqing Xie,[1] **Fan Dong**,[1] **and Xia Feng**[2]

[1]*School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China*
[2]*School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China*

Correspondence should be addressed to Qingqing Xie; xieqq@ujs.edu.cn

Received 19 February 2021; Revised 6 April 2021; Accepted 16 April 2021; Published 28 April 2021

Academic Editor: Lu Liu

The blockchain technology achieves security by sacrificing prohibitive storage and computation resources. However, in mobile systems, the mobile devices usually offer weak computation and storage resources. It prohibits the wide application of the blockchain technology. Edge computing appears with strong resources and inherent decentralization, which can provide a natural solution to overcoming the resource-insufficiency problem. However, applying edge computing directly can only relieve some storage and computation pressure. There are some other open problems, such as improving confirmation latency, throughput, and regulation. To this end, we propose an edge-computing-based lightweight blockchain framework (ECLB) for mobile systems. This paper introduces a novel set of ledger structures and designs a transaction consensus protocol to achieve superior performance. Moreover, considering the permissioned blockchain setting, we specifically utilize some cryptographic methods to design a pluggable transaction regulation module. Finally, our security analysis and performance evaluation show that ECLB can retain the security of Bitcoin-like blockchain and better performance of ledger storage cost in mobile devices, block mining computation cost, throughput, transaction confirmation latency, and transaction regulation cost.

1. Introduction

Since Satoshi Nakamoto invented Bitcoin in 2008 [1], the blockchain technology has gained considerable interest and adoption in multiple fields, such as economics, cryptography, and mathematics. Blockchain makes it possible to process the online trade among mutual distrust parties. The security of the blockchain technology is achieved by sacrificing prohibitive computation and storage resources to jointly maintain a unique transaction ledger. However, most mobile systems are underresourced due to weak mobile devices. As a result, it is a matter of great difficulty to apply the blockchain technology to mobile systems. Some details are shown as follows:

On the one hand, each miner contributes immense computation effort to painstakingly solve a cryptographic problem, i.e., the proof of work (PoW) problem. Only the miner who first succeeds in solving the PoW problem can pack some transactions into a new valid block and append to the longest ledger. Generally, the mining machines, such as ANTMINER S9 Hydro, reach up to 18TH/s [2], while the hash rates of the normal mobile devices are just at the MH/s level. According to the statistics, Bitcoin alone is estimated to use tens of Terawatt hours per year, which is enough to power a mid-sized country. It indicates that most normal mobile devices cannot undertake the mining work because of their limited computational power.

On the other hand, each miner has to maintain an entire copy of the transaction ledger, i.e., every transaction record from the beginning of time. Storing the entire blockchain ledger requires a remarkable amount of storage capacity. Take Bitcoin as an example; the total size of a local ledger reached more than 380 GB on February 18, 2021 [3]. And, it is growing at a rate of around 70 MB per day. It is not feasible

for a normal mobile device to store such a large-size ledger. Table 1 shows some ledger growth information in several different blockchain systems.

In conclusion, most mobile devices are unable to provide such computation and storage capacities to meet the requirements for working as miners. The aforementioned issues must be solved to popularize the blockchain applications in mobile systems.

Edge computing appears with inherent decentralization and strong resources, which can provide a natural solution to overcoming the aforementioned resource-insufficiency situation [7, 8]. As Abbas et al. pointed out in [9], edge computing is now a promising technology in the 5G mobile environment. Each edge node locates close to the end devices at the edge network and can provide sufficient capacities of storage, computation, and networking to support the mining work. That is, edge computing relies on edge nodes to create services that are distributed across edge domains. Thus, constructing a lightweight blockchain system based on edge computing is a natural and appropriate way to make the blockchain technology widely used in practical mobile systems.

1.1. Challenge. The direct integration of blockchain and edge computing can only relieve some storage and computing pressure at end mobile devices. But first, the end devices do not always work as completely light nodes. They usually are interested in some types of transaction information, maybe related to their jobs, life, or something else. They often want to store some transactions as well. Second, there are some other open problems, such as improving the transaction confirmation latency, throughput, etc. These performance metrics must be optimized when applying the blockchain technology to mobile systems. It is a paradox. The reason is that, on the one hand, the high transaction confirmation latency and low throughput are caused by the computation-intensive consensus protocol itself. On the other hand, the computation-intensive consensus protocol is a key to maintaining the security and stability of blockchain systems.

To sum up, the challenge is how to solve this paradox to achieve both light weight at end mobile devices and superior performance regarding transaction confirmation latency and throughput.

Some related works have been done so far. Cebe et al. [10] proposed an integrated lightweight blockchain framework for forensic applications of connected vehicles, abbreviated as Block4Forensic. In Block4Forensic, each node maintains a shared ledger and a fragmented ledger. The shared ledger stores hash values. The fragmented ledger stores some transactions attracted by the corresponding participant. Liu et al. [11] proposed a mobile edge-computing-enabled wireless blockchain framework where the computation-intensive mining tasks could be offloaded to the nearby edge nodes and the cryptographic hashes of blocks could be cached in the edge servers. Chen et al. [12] proposed a multi-hop cooperative and distributed computation offloading algorithm that considered the data processing tasks and the mining tasks together for blockchain-empowered Industrial Internet of Things (IIoT). Lei et al. [13] proposed Groupchain, a novel scalable public blockchain of a two-chain structure suitable for fog computing. To some extent, Groupchain overcomes the scalability challenge of blockchain's integration with fog computing. Eyal et al. [14] proposed Bitcoin-NG, a Byzantine fault-tolerant blockchain protocol. It decouples Bitcoin's blockchain operation into leader election and transaction serialization. It introduces high generation frequency of micro-blocks for transaction commitment. Table 2 shows the advantages and disadvantages of these works. These works give us great inspiration to study the blockchain application problems. There are also some other related works [15–17]. All these works can only solve part of the aforementioned challenges.

Table 1: Ledger growth information in several different blockchain systems, according to the statistics on February 18, 2021 [3].

Blockchain	Block interval	Block count	Ledger size (GB)
Bitcoin [1]	10 m·45 s	671,200	382.04
Ethereum [4]	13.3 s	11,884,711	608.20
Bitcoin cash [5]	10 m·40 s	675,436	184.14
Litecoin [6]	2 m·28 s	2,003,322	40.64

1.2. Contributions. The main contributions are summarized as follows:

(1) We propose a novel lightweight blockchain framework based on edge computing (ECLB) for mobile systems. It takes edge nodes as miners, to relieve some storage and computation pressure at end mobile devices. As for the mobile devices, we introduce the fragmented ledger structure [10], to let them obtain the transaction information of interest. In this proposed ECLB framework, edge computing and blockchain technology complement each other, which makes the blockchain technology applicable in mobile systems.

(2) Under the ECLB framework, we reform the block structures into leader block and transaction block. The leader blocks are used to record leader nodes, who succeed in solving the PoW puzzles. The transaction blocks are used to record the transaction history via most edge nodes' signature assurance. Such a structure optimizes the blockchain metrics, including throughput and transaction confirmation latency.

(3) Considering the popular permissioned blockchain settings, we specifically utilize symmetric encryption algorithm and ciphertext-policy attribute-based encryption (CP-ABE) scheme [18] to design a pluggable regulation layer. It is a secure solution for supervising the transaction behaviors. Note that due to the low efficiency, the CP-ABE schemes cannot be readily adopted. Here, in order to meet the requirement of high efficiency, we combine CP-ABE with symmetric encryption algorithm to improve the regulation efficiency.

Table 2: Advantages and disadvantages of some existing works.

Research	Advantages	Disadvantages
[10]	Make each end device to maintain a fragmented ledger, to reduce the storage pressure	Does not consider improving the transaction throughput and confirmation latency
[11]	Offload the computation-intensive mining tasks to nearby edge-computing nodes	
[12]	Disburden the data processing tasks and mining tasks from end devices to edge servers	
[13]	Employ a leader group to optimize the transaction throughput and confirmation latency	Does not consider reducing the ledger storage pressure at end devices
[14]	Decouple Bitcoin's blockchain operation into leader election and transaction serialization to achieve scalability	

(4) We analyze the security to demonstrate that our ECLB achieves fault tolerance, high security level with 16 edge nodes, Sybil attack resistance, double-spending attacks resistance, and chosen-plaintext attack (CPA) resistance. We also conduct performance evaluation, demonstrating that ECLB achieves lower cost of ledger storage and block mining computation, and better throughput, transaction confirmation latency, and regulation efficiency.

1.3. Structure of the Paper. The rest of the paper is organized as follows: Section 2 presents the preliminaries. Section 3 presents our ECLB system model. Section 4 presents our ECLB protocol design. Sections 5 and 6 formally analyze the security and experimentally evaluate the performance of our ECLB. Section 7 reviews the related works. Section 8 discusses the solution to the blockchain fork problems. Finally, Section 9 concludes this paper.

2. Preliminaries

We briefly review the blockchain technology and the CP-ABE scheme.

2.1. Blockchain. Blockchain was invented by Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the Bitcoin cryptocurrency [1]. The ledger records a continuously growing list of transactions, called blocks, which are linked by the cryptographic hash of the previous block. The general structure of the block and the blockchain is shown in Figure 1.

A blockchain is typically managed by a peer-to-peer (P2P) network collectively following a predefined consensus protocol. Each miner contributes a large amount of computation energy for packing transactions into a new block, i.e., the consensus procedure or mining tasks. As we know, PoW is a frequently and widely used consensus protocol, such as in the Bitcoin systems. PoW requires a complicated computational process for packing transactions. It is a random process where a lot of trials and errors are required on average before a PoW solution is generated. In PoW, all the miners have to use different nonces and calculate the hash value of the constantly changing block header continuously, until the calculated hash value is not greater than a given value. When one node obtains the target, all other nodes must mutually confirm the correctness of the value. Finally, a new block is generated. The flow of new block generation procedures is shown in Figure 2. A new block is determined in a round.

The characteristics of the blockchain technology are listed as follows:

(i) Decentralization: the blockchain is built on a P2P network, which is naturally decentralized. All participating nodes have the same copy of the blockchain ledger.

(ii) Immutability: once a block is written to a blockchain, the information cannot be altered.

(iii) Authenticity: users can trust that transactions will be executed exactly as the protocol comments. Thus, the transaction data in blockchain ledger are all authentic.

(iv) Pseudonymity: blockchain uses a pseudo-identity mechanism. Each user can generate as many pseudo-identities as he/she likes to increase identity privacy.

Obviously, it should reduce the pressure of both ledger storage and block mining computation to design a thoughtful lightweight blockchain system. Simultaneously, the scalability is also an important factor to measure a blockchain system. Scalability itself includes two important metrics: throughput and transaction confirmation latency.

2.2. CP-ABE. The CP-ABE scheme was proposed to achieve fine-grained access control [18]. In CP-ABE, a user's secret attribute key is associated with an attribute set. The ciphertext of a message is associated with an access policy. A user will succeed in decrypting a ciphertext if and only if the user's attribute set matches the access policy associated with the ciphertext.

The CP-ABE scheme consists of four algorithms [18]:

(i) $\text{CPABE.Setup}(1^\lambda) \longrightarrow \text{MK}, \text{PK}$: it takes the security parameter 1^λ as input, and outputs a master key MK and a public key PK.

(ii) $\text{CPABE.Encrypt}(T, m, \text{PK}) \longrightarrow \text{CT}$: it takes as input an access policy tree T over the universe of attributes, a message m, and the public key PK, then encrypts m

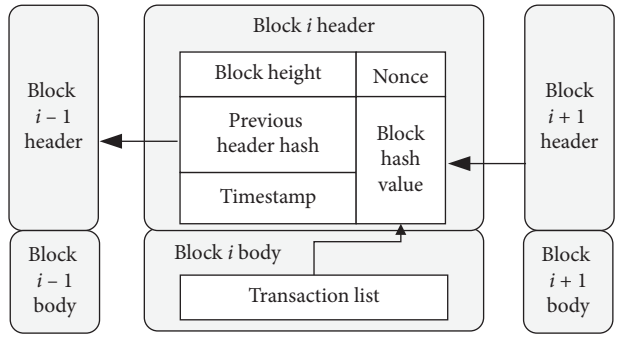

Figure 1: The chain structure of the blockchain.

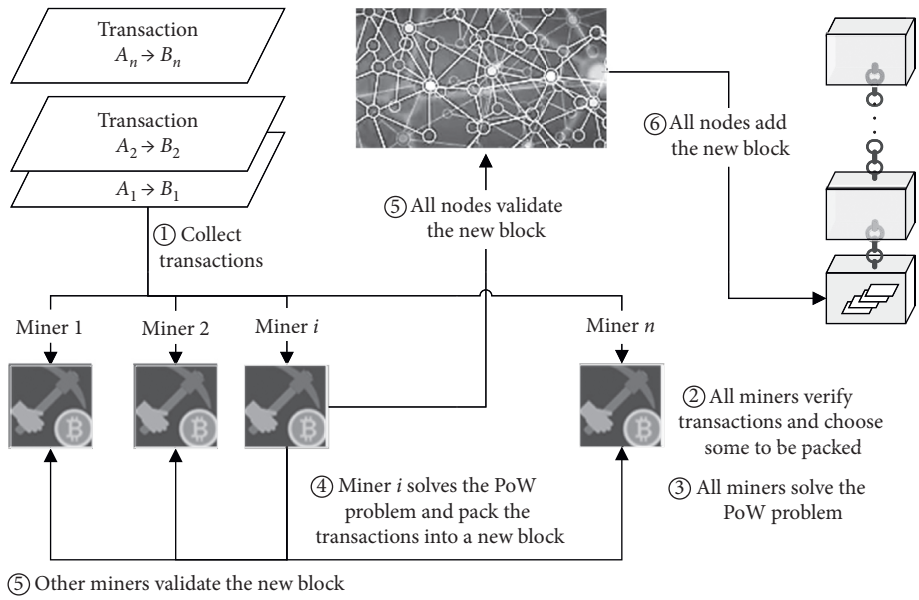

Figure 2: The flowchart of new block generation procedures.

as a ciphertext CT, such that only a user that possesses a set of attributes that satisfies the access tree T will be able to decrypt the message m.

(iii)CPABE.KeyGenerate$(\mathrm{PK}, \mathrm{MK}, A_u) \longrightarrow \mathrm{SK}_u$: it takes as input the public key PK, the master key MK, and a user's attribute set A_u, then outputs the user's secret attribute key SK_u.

(iv)CPABE.Decrypt$(\mathrm{PK}, \mathrm{CT}, \mathrm{SK}_u) \longrightarrow m$: it takes as input the public key PK, the ciphertext CT of a message m, and a user's secret attribute key SK_u, then outputs the message m if the user's attribute set satisfies the access policy tree associated with CT.

3. System Model

The conception model of our ECLB framework is shown in Figure 3. It mainly consists of the following four layers:

(1) Cloud data center layer: it is in charge of storing encrypted transaction information specifically for the permissioned blockchain setting. We assume that the cloud data center is honest but curious. That means, it acts in an honest fashion and correctly follows the designated protocol specification. However, it is curious to infer and analyze the stored data to harvest additional information to gain illegal profits.

(2) Edge nodes layer: each node on this layer undertakes the mining work as a blockchain miner node, i.e., solving the PoW puzzles and storing an entire copy of the blockchain ledger. Each edge node i has a public/private key-pair (pk_i, sk_i). It is either honest or Byzantine. Byzantine nodes do not follow the consensus protocol accidentally or maliciously. It means that they might fail to join the consensus or collude to attack the whole network. Assume that there are n edge nodes, these n nodes are well connected in a P2P network, and the number of Byzantine nodes is f, where $n \geq 3f + 1$ is required in our model [13].

(3) End devices layer: it consists of some traditional PC or mobile computing end devices, such as laptop, smart phone, etc. They usually provide weak capacities of computing, storage, and networking.

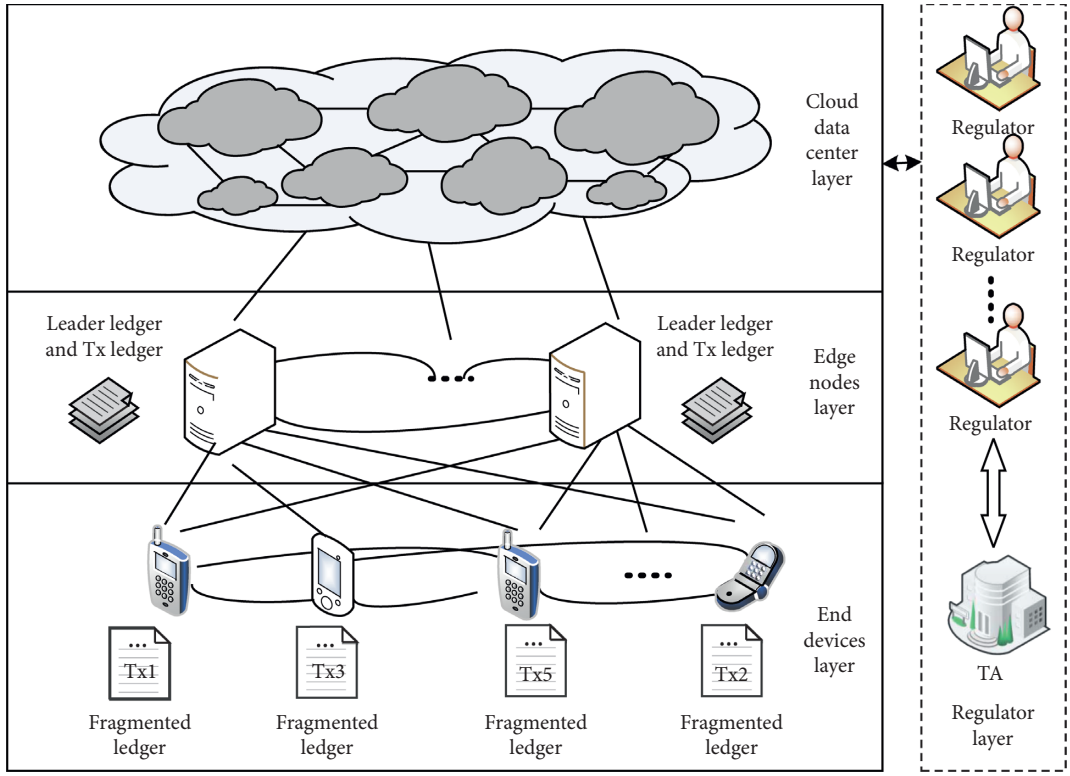

Figure 3: Our ECLB model.

Hence each end device only stores a fragmented ledger [10], consisting of the copy of the block headers and some transaction records of interest. End devices are usually too weak to be miners. They only download some transaction information of interest from nearby edge nodes. Thus, they can be trustworthy or not, which has no effect to the whole network.

(4) Regulator layer: it consists of some regulators and a trusted authority (TA). This layer is designed specifically for the permissioned blockchain setting. On the one hand, the regulators request to gain the transaction data in cloud to carry out trading regulation. On the other hand, considering the transaction privacy preservation, only the regulators are allowed to get the transaction data. And, they are not allowed to get the data outside of their privileges. Thus, the regulators are assumed to be honest but curious. The TA is in charge of controlling the access privilege, i.e., authorizing the access privilege only to the regulators. The TA is assumed to be trustworthy.

4. ECLB Protocol Design

In this section, we will describe our ECLB protocol design in detail, including three parts: transaction ledger storage, transaction packing and confirmation, and transaction regulation. Some major notations used in our ECLB protocol are shown in Table 3.

Table 3: Some major notations used in our ECLB protocol.

Notations	Description
Node_i	The i-th edge node
(pk_i, sk_i)	The public and private key pair of Node_i
n	The number of edge nodes
b_t	The candidate transaction block generated at time t
$\text{Sign}(\cdot,\cdot)$	A signing algorithm
$\text{Verify}(\cdot,\cdot)$	A signature verification algorithm corresponding to $\text{Sign}(\cdot,\cdot)$
s_t^i	Node_i's signature on b_t, i.e., $s_t^i = \text{Sign}(sk_i, b_t)$
UTXO	The unspent transaction output (UTXO) set
vr_t	The verification result of the candidate block b_t
APT	An access policy tree
PK	A public key in the regulator layer
MK	A master key of the trusted authority in the regulator layer
key	A symmetric key
CT_{key}	The ciphertext of key
$\text{SE}(\cdot,\cdot)$	A symmetric encryption algorithm
$\text{SD}(\cdot,\cdot)$	A symmetric decryption algorithm
tx_i	The i-th transaction record
CTX_i	The ciphertext of the transaction record tx_i

4.1. Transaction Ledger Storage. In real applications, the edge nodes are located close to the end mobile devices, and have much stronger storage and computation capabilities compared with the end mobile devices. Thus, we take the edge nodes as blockchain miners and the edge devices as light nodes.

Specifically, in our framework, there are two chains: a leader chain and a transaction (Tx) chain. There are two kinds of blockchain ledgers: full ledger and fragmented

ledger. The full ledger records the identities of both the leaders and the transaction history, by packing the public keys of the leaders and the transaction records. The fragmented ledger records the block headers of the full ledger and some transaction records attracted to the corresponding end mobile devices. Obviously, the fragmented ledger [10] is specifically introduced for the end devices. Each edge node stores an entire copy of the full ledger. The structure of the transaction ledger storage is described in Table 4.

As their name imply, the *leader chain* packs the public keys of the leader, while the *Tx chain* packs the whole transaction records. The ledgers produced by both the leader chain and the Tx chain form the *full ledger*. All the block headers in the full ledger and a part of the transactions packed by the Tx chain form the *fragmented ledger*. Obviously, the size of the fragmented ledger is much smaller than that of the full ledger. The *edge nodes* play the role of *miner nodes*, and thus are responsible to store the full ledger. The *end mobile devices* only need to store the fragmented ledger, due to their weak resources. Thus, they play the role of *light nodes*. Nevertheless, they still can obtain the transaction information since the fragmented ledger maintains a part of transactions.

4.2. Transaction Packing and Confirmation. Section 4.1 introduces lightweight ledger storage at end mobile devices. In this part, we will describe the scalability optimization and lightweight mining computation.

Inspired by [13, 14], we construct a leader group to achieve high scalability. The edge nodes participating in the mining work form the leader group. Assume that there are n edge nodes (i.e., miners) that collectively commit transactions via new blocks, and at most $(n-1)/3$ of them are Byzantine nodes.

In our ECLB, there are two chains growing in parallel: a leader chain and a transaction (Tx) chain, as shown in Figure 4. For convenience, we simply call the blocks in the leader chain *leader blocks*, and the blocks in the Tx chain *Tx blocks*. The leader chain is used to record which edge node competes successfully for serving as a leader.

In our ECLB, first each edge node tries to solve a PoW problem to mine a leader block for competing for being a leader. The leader block packs its own public keys and the corresponding reward coinbase. Once an edge node wins, denoted as Node_i, a new leader block will be generated and broadcast to all the other edge nodes. Node_i chooses and packs some new transactions into a Tx block by embedding its signature as assurance. In parallel, all the edge nodes still can work on solving another PoW problem to compete for being a leader. Once another edge node wins, denoted as Node_j, Node_j will be a new leader and the aforementioned procedures are repeated. Note that an edge node can be a leader in succession, i.e., $i = j$ may happen. We can see that only the leader has the right to pack new transactions.

Now we present the aforementioned transaction packing and confirmation process, as follows:

(1) To compete for being a leader, each edge node works on mining a leader block by solving a PoW problem. Once an edge node succeeds in solving the PoW and gets a valid leader block, it immediately broadcasts the leader block to all the other edge nodes. Assume that Node_i is the winner. All the other nodes check its validity and append the leader block to the local leader chain if it is valid. In parallel, all the edge nodes still can work on mining a new leader block based on the latest leader chain, for replacing the original leader.

(2) The leader, i.e., the winning edge node Node_i, first packs a set of new transactions, then computes a signature s_t^i, and finally generates a corresponding new candidate Tx block b_t and broadcasts b_t to the other edge nodes. The candidate Tx block generation algorithm is shown in Algorithm 1.

(3) Once receiving a candidate Tx block b_t generated by the leader, each other edge node Node_j verifies b_t based on the signatures and UTXOs, where $j = 1, 2, \ldots, i-1, i+1, \ldots, n$. The verification algorithm of the candidate Tx block is shown in Algorithm 2. If Node_j verifies that the candidate Tx block is valid, it will sign the candidate block b_t as $s_t^j = \text{Sign}(sk_j, \text{hash}(b_t))$, and broadcast s_t^j to other nodes.

(4) All the edge nodes collect the signed block from each other edge node. If an edge node obtains the signed block b_t from 2/3 supermajority, meaning that all the edge nodes agree on the candidate block b_t, then b_t will be appended to the Tx chain.

(5) Repeating steps (2)–(4) until another leader block is generated. That is, during the steps (2)–(4), in parallel, all the edge nodes work on solving a PoW problem and mining a new leader block to compete for being a leader.

The transaction packing and confirmation processes are shown in Figure 5, assuming that the edge node Node_0 is the leader who is the first to succeed in solving PoW, Node_3 is faulty.

4.3. Transaction Regulation. In public/permissionless blockchain systems, any transaction information is available to any entity in the network, which provides much convenience to the regulator department. However, in the permissioned blockchain, only the blockchain member nodes are allowed to obtain the transaction information. Hence, an interface of reading the Tx ledger needs to be set for outside regulator department. To this end, we will design a transaction regulation protocol specifically for the permissioned blockchain setting.

Considering the requirements of both privacy preservation and secure regulation, we will employ the CP-ABE scheme to realize secure sharing of the transaction records with legal regulators. However, the CP-ABE scheme is

Table 4: Transaction ledger storage at edge nodes and end devices.

Roles	Node types	Ledger types	Ledger contents
Miner nodes	Edge nodes	Full ledger	The public keys of the leaders and the whole transaction records
Light nodes	End mobile devices	Fragmented ledger	The block headers and some transactions of interest

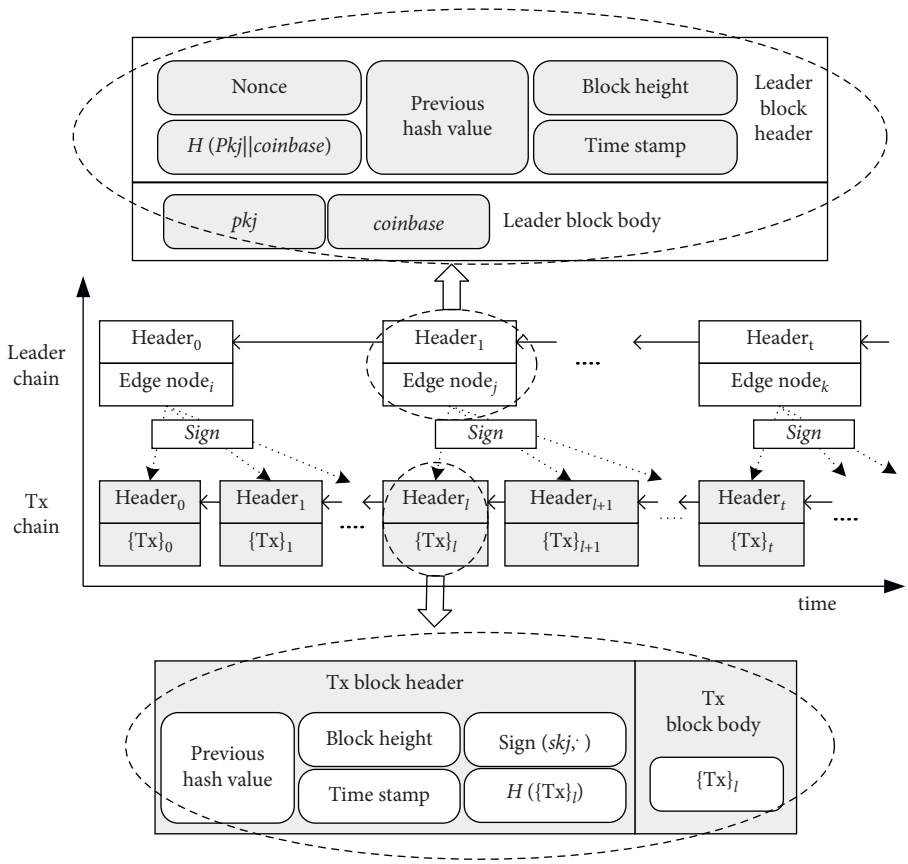

Figure 4: Two-chain structure of ECLB.

Input:
$\{Tx\}_{new}$: The set of new transactions;
PreHash: Hash of previous block header;
t: Timestamp;
sk_i: Secret key of the leader;
Output:
b_t: Candidate block;
(1) Select a set of valid transactions from $\{\text{Tx}\}_{new}$, denoted as $\{Tx\}_t$;
(2) Set body←$\{\text{Tx}\}_t$;
(3) Construct a Merkle hash tree MT over $\{\text{Tx}\}_t$, and denote its root hash as *NowHash*;
(4) Increase blockHeight by blockHeight←blockHeight + 1, where blockHeight represents the block height and sets zero in the genesis block;
(5) Compute a signature $s_t^i = \text{Sign}(sk\_i, \text{hash}(\text{blockHeight}\|t\|\text{PreHash}\|\text{NowHash}))$;
(6) Set header←{blockHeight, t, PreHash, NowHash, s_t^i};
(7) Set b_t←header, body;
(8) return b_t;

Algorithm 1: Candidate Tx block generation.

Input:
b_t: A candidate Tx block;
$UTXO$: The UTXO set;
pk_i: The public key of the current leader node;
Output:
vr_t: Verification result;
(1) Initialize $vr_t \leftarrow valid$;
(2) Obtain the leader's signature s_t^i from b_t;
(3) Obtain the $blockHeight||t||PreHash||NowHash$ from b_t and compute $h = hash(blockHeight||t||PreHash||NowHash)$;
(4) Verify the signature $s_t^i$by computing $h' \leftarrow Verify(pk_i, s_t^i)$;
(5) if $h' \neq h$ then
(6) Set $vr_t \leftarrow invalid$;
(7) return vr_t;
(8) end if
(9) Get the transaction set from b_t, denoted as $\{Tx\}_t$;
(10) for $tx \in \{Tx\}_t$ then
(11) Check validity of tx based on $UTXO$;
(12) if tx is invalid then
(13) Set $vr_t \leftarrow invalid$;
(14) return vr_t;
(15) end if
(16) end for
(17) return vr_t;

Algorithm 2: Candidate transaction block verification.

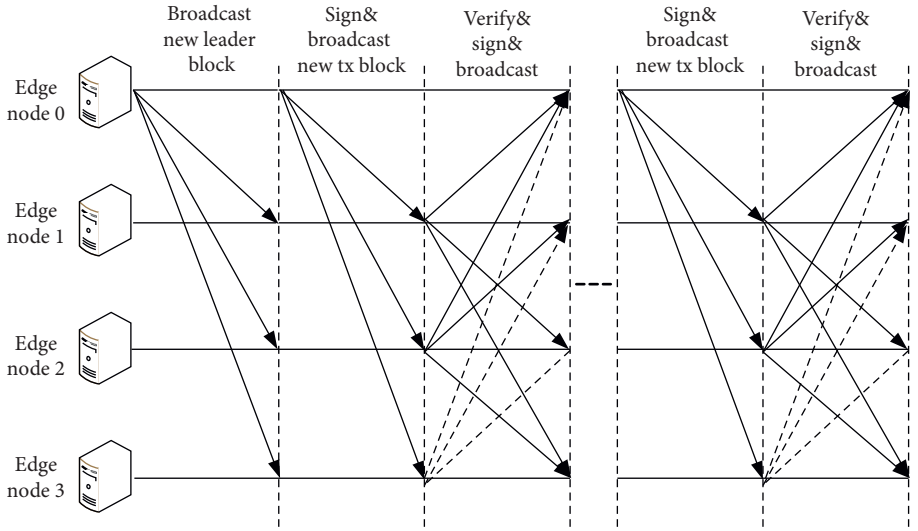

Figure 5: The flowchart of the transaction packing and confirmation protocol.

notoriously inefficient in encryption and decryption. To solve this problem, we will utilize the key encapsulation mechanism to improve the efficiency [19, 20]. First, the central control of the permissioned blockchain encrypts a symmetric key key using the CP-ABE scheme. Then, key is shared with all the edge node members and repeatedly used to encrypt the valid and newly packed transaction records. Last, only the designated data consumers, i.e., valid regulators, can succeed in decrypting the key, and further decrypting the transaction records by the key. As a result, the transaction records are stored in ciphertext format in cloud server and can only be accessed by the legal regulators. The detailed transaction regulation protocol consists of the following steps:

(1) The central controller of the permissioned blockchain generates a symmetric key key and determines an access policy tree APT. Then, it calls the CPABE.Encrypt algorithm to encrypt key under APT, as

$$CT_{key} = CPABE.Encrypt(APT, key, PK) \quad (1)$$

CT_{key} is outsourced to the cloud for storage. In addition, the central controller sends the symmetric key key to all the edge nodes.

(2) A regulator requests a secret attribute key SK_u from the trusted authority (TA) in the regulator layer. The TA calls the CP-ABE's key generation algorithm to compute

$$SK_u = \text{CPABE.KeyGenerate}(PK, MK, A_u) \quad (2)$$

where A_u is the regulator's attribute set. SK_u is sent to the corresponding regulator.

(3) The regulator downloads the key ciphertext CT_{key} from the cloud, and uses his or her secret attribute key SK_u to decrypt the symmetric key key, i.e.,

$$key = \text{CPABE.Decrypt}(PK, CT_{key}, SK_u) \quad (3)$$

If his or her attribute set A_u satisfies the access policy tree APT, he or she will obtain key, otherwise null.

(4) Once a new Tx block b_t is committed, the corresponding leader uses the symmetric key key to symmetrically encrypt each transaction record tx_i of b_t as

$$CTX_i = SE(key, tx_i), \quad (4)$$

where $SE(\cdot,\cdot)$ represents a symmetric encryption. Each CTX_i is outsourced to the cloud server. This step is repeated with each new committed Tx block.

(5) The regulator downloads the transaction ciphertext CTX_i from the cloud, and symmetrically decrypts it by key to obtain the plain transaction records, i.e.,

$$tx_i = SD(key, CTX_i), \quad (5)$$

where $SD(\cdot,\cdot)$ is the symmetric decryption algorithm corresponding to $SE(\cdot,\cdot)$.

Note that the aforementioned steps (1) and (3) are, respectively, one-time computation during the symmetric key's life cycle. It can be set very long until *key* is leaked. Step (2) is also a one-time computation for each regulator. Hence, the online computation cost of this transaction regulation module mainly depends on steps (4) and (5). These two steps are symmetric encryption and decryption, which are efficient obviously. Thus, the real-time transaction regulation is well supported in our ECLB.

In conclusion, we design an efficient transaction regulation module specifically for the permissioned blockchain setting, by combining the CP-ABE scheme with the key encapsulation mechanism. This transaction regulation module preserves the transaction privacy preservation and simultaneously supports efficient regulation required by the practical government department.

5. Security Analysis

In this section, we will provide some security analysis, including fault tolerance, the least number of edge nodes to reach a high security level, Sybil attack, double-spending attack, and chosen-plaintext attack (CPA).

5.1. Fault Tolerance. The security of fault tolerance is analyzed by proving the following theorem.

Theorem 1. *The edge nodes guarantee fault tolerance, if the number of Byzantine edge nodes f is no more than $(n-1)/3$, i.e., $n \geq 3f+1$, where n is the total number of edge nodes.*

Proof. Assume all the edge nodes are divided into three disjoint sets, i.e., H_1, H_2, B, where H_1 and H_2 represent two sets of honest edge nodes and B are all Byzantine nodes. Thus, we have

$$|H_1| + |H_2| + |B| = n, \quad (6)$$

and for the worst case,

$$|B| = f. \quad (7)$$

If the Byzantine edge nodes in B want to change the system status, they need to first mine a leader block to propose a consensus process. In this way, malicious nodes can gain agreement from supermajority edge nodes. To win this attack, it requires

$$|H_1| + |B| \geq n - f, \quad (8)$$

$$|H_2| + |B| \geq n - f. \quad (9)$$

By simplifying equations (6)–(9), we can get

$$f \geq n/3. \quad (10)$$

Therefore, all the edge nodes are able to guarantee fault tolerance if the number of Byzantine members f is no more than $(n-1)/3$, i.e., $n \geq 3f+1$. □

5.2. The Number of Edge Nodes. We assume that each edge node is either honest or Byzantine, and the mining is a fair game. Let p be the probability of that an edge node is Byzantine. As mentioned in Section 5.1, there are less than $f = (n-1)/3$ Byzantine edge nodes. Thus, using the cumulative binomial distribution, the security probability of the leader chain is computed as

$$P[X \leq f] = \sum_{k=0}^{f} \binom{n}{k} p^k (1-p)^{n-k}. \quad (11)$$

Considering that in the Bitcoin, the recommended 6-block-confirmation is calculated under $p = 0.1$ and security level ≥0.99, we will set the same adversary probability and security level. The leader chain ensures the same security as long as there are not less than 16 edge nodes, as shown in Figure 6. It means that as long as our ECLB framework is configured with no less than 16 edge nodes, the security level ≥0.99 can be guaranteed.

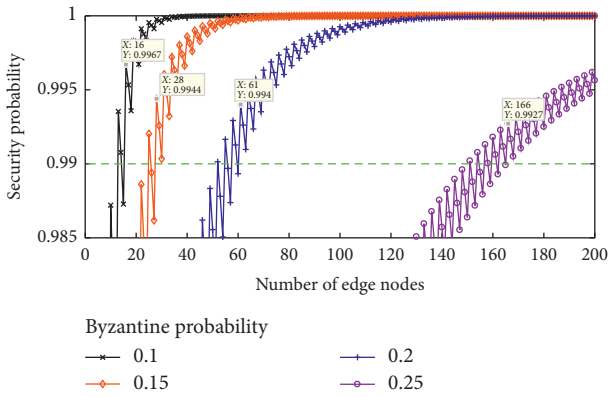

Figure 6: Security under different byzantine probabilities.

5.3. Sybil Attack. Sybil attacks [21] allow a malicious participant to subvert a peer-to-peer network by creating many pseudonymous identities in order to work as multiple distinct nodes.

By using PoW to compete for being a leader, the leader chain has a natural ability to resist Sybil attacks. Recall that once an edge node becomes a leader, it is the only one to be allowed to broadcast blocks. In order to become a leader, it must solve a PoW problem, which is extremely computationally intensive. PoW raises the cost of creating a new leader identity. Thus, it mitigates Sybil attacks, wherein security property is guaranteed by the leader chain.

5.4. Double-Spending Attacks. In the leader chain, any edge node checks the collective signatures of a Tx block, in which a supermajority (i.e.1/2) of the edge nodes permit its validity. In other words, the Tx chain is under the supervision of all the edge nodes instead of a single leader. Thus, a double-spending attacker will have no chance to use the same coin(s) to issue two (or more) transactions [22]. Moreover, in this respect, 0-block-confirmation services can be provided for clients in a secure way.

5.5. Chosen-Plaintext Attack (CPA). We first give Definition 1 of CPA security of our ECLB protocol. We then demonstrate the CPA security of our ECLB protocol by proving Theorem 2.

Definition 1. Our ECLB protocol is CPA secure if the transaction regulation protocol is CPA secure.

Theorem 1. *Our ECLB protocol is CPA secure.*

Proof. We reduce the CPA security proof of our ECLB protocol to that of the transaction regulation protocol. As we know, there are some efficient and symmetric encryption algorithms that are secure against CPA, such as AES and DES. Hence, whether the transaction regulation protocol is secure against CPA depends on the indistinguishability of the symmetric key's ciphertext against CPA. The indistinguishability of the symmetric key key's ciphertext is guaranteed by the CPA security of the traditional CP-ABE scheme [18]. Thus, it proves that the transaction regulation protocol is secure against CPA. Finally, according to Definition 1, our ECLB protocol is also CPA secure. □

6. Performance Evaluation

6.1. Implementation. We extend the Bitcoin Simulator [23] to implement the key elements of the transaction packing and confirmation process for performance analysis, with the absence of Byzantine nodes. The transaction regulation protocol is implemented using the Java Pairing-based Cryptography (JPBC) library [24]. The experimental machine is configured with Intel(R) CORE(TM)2 Duo CPU E8400 @ 3.00 GHz and 8.00 G RAM. In addition, we simulate the broadcast, sign, and verify procedures by imposing a latency of 100 ms for each edge node [13]. The reason is that the network topology is almost a complete graph, and the broadcast procedure is very fast.

6.2. Ledger Storage. We set the size per transaction at around 256 bytes, and the size per block at 1 MB. Thus, one block contains around 4000 transactions. In Bitcoin, each full node, i.e., miner, stores the entire transaction ledger, while each light node stores only the block headers. In ECLB, each edge node stores the entire leader ledger and transaction ledger, while each end mobile device stores the fragmented ledger, i.e., only all the block headers and some transaction of interest. Note that the leader ledger is very small compared with the transaction ledger, since only one leader block is mined after around every 1500 transaction blocks. Hence, we speculate that the ledger storage cost at an edge node in ECLB is almost as high as that at a full node in Bitcoin. The ledger storage cost at the end mobile device in ECLB will be slightly higher than that at the light node in Bitcoin but much lower than that at the full node and the edge node.

Figure 7 actually demonstrates the aforementioned speculation, where "x% ToI" represents an average x% percentage of transactions stored by each end mobile device. The ledger storage cost at light node of Bitcoin is too small to be shown. Even though the ledger storage cost at end devices is slightly higher than that at light node in Bitcoin, for 3.6×10^6 transactions, it costs around 86 MB to store the fragmented ledger at the end device with 10% transaction of interest. Thus, our ECLB achieves lightweight ledger storage at end mobile devices.

6.3. Block Mining. In Bitcoin, each miner needs to solve a PoW problem for mining a new block. While in our ECLB, only the leader block is mined through solving a PoW problem. All the Tx blocks are created by only the corresponding signatures, which is much lighter than solving a PoW problem. Most importantly, the leader block mining and the Tx block creation procedures are executed in parallel. Thus, our ECLB holds lightweight and efficient block mining process. Figure 8 shows the block mining time with

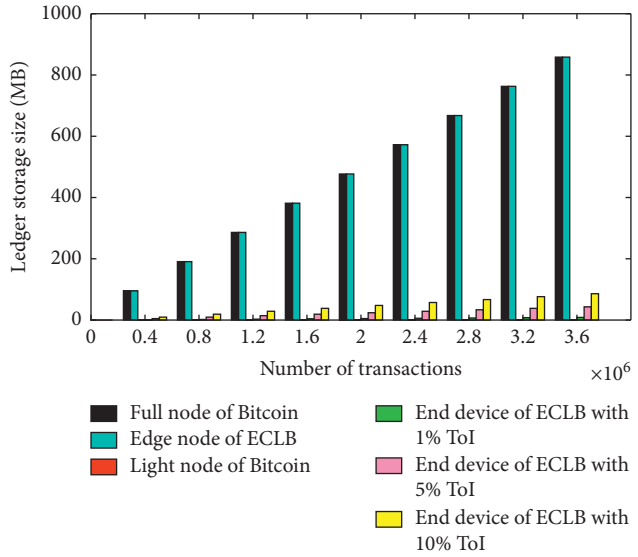

Figure 7: Ledger storage size with different number of transactions to pack.

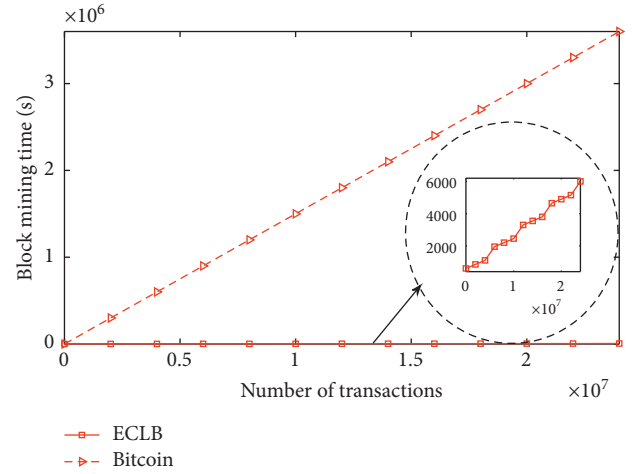

Figure 8: Block mining time with different number of transactions.

different number of transactions. It sufficiently demonstrates that our ECLB provides lightweight and efficient block mining.

6.4. Throughput. We set block frequency to 1 per 10 minutes for Bitcoin and the leader block frequency as the same. Obviously, the throughput of our ECLB is shown by only the Tx chain. We test the throughputs with different block sizes. Figure 9 shows the experimental results. We observe that our ECLB achieves much higher throughput than Bitcoin of 100 times on average.

6.5. Transaction Consensus Latency. Since the transaction commitment is submitted through the Tx chain, we only consider the transaction block commitment among the edge nodes for the transaction consensus latency. To see the scalability of ECLB's consensus process in terms of the number of edge nodes, we set the transaction block size to 1 MB, which is the maximum block size in current Bitcoin. In Bitcoin, the consensus latency is the time for at least 50% nodes to receive a block. Groupchain [13] and our ECLB have 3 and 2 rounds of interactions on average, respectively. Figure 10 shows the experimental results. We observe that our transaction consensus latency is slightly higher than that of Bitcoin and lower than that of Groupchain. But Groupchain and our ECLB allow the blocks already appended to the blockchain to be confirmed valid immediately without the 6-block confirmation, while the Bitcoin needs 6-block-confirmation mechanism.

6.6. Regulation Efficiency. We evaluate the regulation efficiency from the aspects of online transaction encryption and decryption, i.e., Steps (4) and (5) in Section 4.3. The reason is

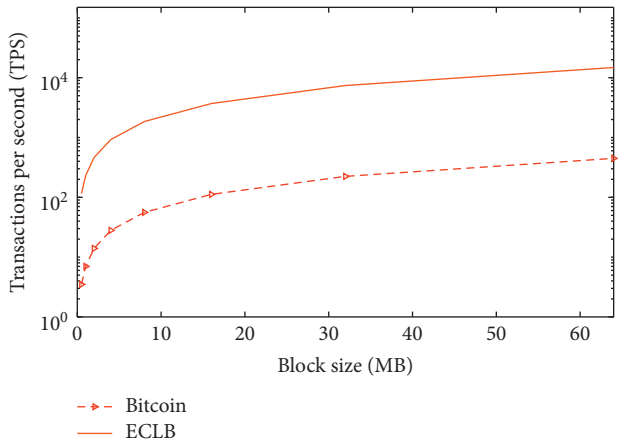

Figure 9: Tx throughput with different block sizes.

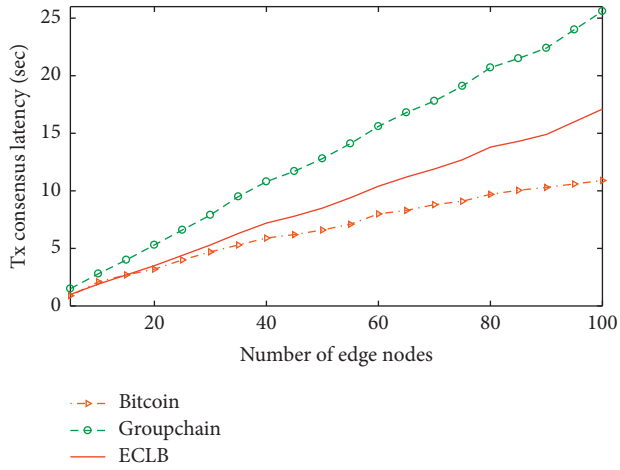

Figure 10: Transaction consensus latency with different number of edge nodes.

that the other 3 steps need to run only once and can be performed offline and in advance. Figure 11 shows the online encryption and decryption time cost. We observe that they are constant and low enough to satisfy efficient transaction regulation.

7. Related Works

In this section, we introduce some related works in the area of lightweight blockchain and access control.

7.1. Lightweight Blockchain. Since the advent of blockchain technology, much effort has been devoted to designing lightweight blockchain systems for decentralized Internet of Things [25]. Liu et al. [15] proposed a lightweight blockchain system to alleviate the resource occupation of blockchain and made it suitable for IIoT. Specifically, the work exploited an Unrelated Block Offloading Filter (UBOF) to detect and offload unrelated transactions, thus achieving lightweight feature. However, offloading "unrelated transactions" will hinder the transaction regulation in the future. For long-term consideration, all the transaction records should be stored completely. Qu et al. [26] proposed a lightweight blockchain model based on hypergraphs. They used the hypergraph theory to partition the entire network into many hyperedges. Each hyperedge stores a part of transaction data to reduce the storage pressure. However, there are many nodes thus many transaction copies inside the same hyperedge, and one node might belong to more than one hyperedge. But, it brings inconvenience for transaction data sharing. In addition, high data redundancy is still not well-addressed. Cebe et al. [10] proposed an integrated light-weight blockchain framework for forensic applications of connected vehicles. In the work, each participant maintains a shared ledger and a fragmented ledger. The shared ledger keeps only hash values. The fragmented ledger keeps only some information that is of interest to the corresponding

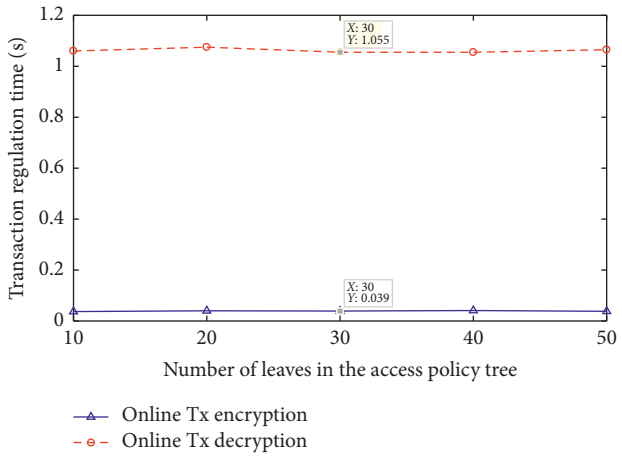

Figure 11: Transaction regulation efficiency with different leaf numbers of an access policy tree.

participants. The fragmented ledger greatly inspired us to design a lightweight ledger storage format at weak end devices. Lei et al. [13] proposed Groupchain, a novel scalable public blockchain of a two-chain structure suitable for fog computing of IoT services computing. Groupchain designed a lightweight transaction confirmation protocol to realize 0-block confirmation.

There are also many other works on lightweight blockchain [15–17, 27–29]. Nevertheless, all these works do not achieve light weight in terms of both ledger storage and block mining computation. A simple comparison is shown in Table 5.

7.2. Access Control. In this section, we will discuss some related works where the access control mechanisms were designed to achieve both privacy preservation and flexible data sharing.

Identity-based encryption enables fine-grained data access control [30–32]. As an advancement, attribute-based encryption (ABE) defines a user identity by his/her attribute set. Sahai and Waters [33] first proposed this method to exert access control over encrypted data. Later, Goyal et al. [34] extended the ABE method to key-policy attribute-based encryption (KP-ABE), by associating a user's secret key with an access policy over attributes. The user can decrypt the ciphertext if and only if the attribute set of the ciphertext satisfies the access policy specified in his/her secret key. The encryptor exerts no control over who has access to the data being encrypted. Bethencourt et al. [18] extended the ABE method to the ciphertext-policy attribute-based encryption (CP-ABE), by associating the ciphertext with an access policy over attributes. A user's secret attribute key is generated from his identity attribute set. The user can decrypt the ciphertext if and only if his/her attribute set satisfies the access policy specified in the ciphertext. The access policy maker is able to decide who should have access to the encrypted data. Currently, many works have been done to devote the ABE method to outsourcing and sharing data securely and flexibly.

Table 5: The comparison of lightweight properties in some lightweight blockchain systems.

Works	Lightweight ledger storage	Lightweight block mining
[10]	✓	✗
[13]	✗	✓
[15]	✓	✗
[16]	✓	✗
[17]	✓	✗
[26]	✓	✗
[27]	✗	✓
[28]	✗	✓
[29]	✓	✗
Our ECLB	✓	✓

Ding et al. [35, 36] proposed a privacy-preserving data processing scheme with flexible access control based on the homomorphic encryption of ABE. It realizes various computations over encrypted data in an efficient way and simultaneously flexibly controls the access to data processing results. Belguith et al. [37] introduced a securely outsourcing multi-authority ABE scheme with policy hidden for the cloud-assisted IoT. Our another work [38] proposed an efficient fine-grained access precision control (FAPC) scheme to achieve secure sharing of the same data, under different precisions with different data users. Deng et al. [20] combined the identity-based encryption and identity-based broadcast encryption mechanisms to propose an identity-based encryption transformation scheme. It supports the encrypted data shared with more people beyond those initially designated by the data owner. Xiong et al. [39] constructed a CP-ABE-based storage model for data storing and secure access in a cloud for IoT applications. It introduces an attribute authority management (AAM) module in the cloud storage system functioning as an agent that provides a user-friendly access control and highly reduces the storage overhead of public keys. Multiple ABE approaches have been proposed to implement secure data outsourcing [40–43], and keyword searching [44–46].

Considering the real-time requirement for transaction regulation, we combine the CP-ABE with the key encapsulation mechanism, to design an efficient transaction regulation protocol.

8. Discussion of Forks

The fork problems are not discussed above. There are two parallel chains in our ECLB, i.e., the leader chain and the Tx chain. Hence, there are two kinds of forks. Now, we will talk about the corresponding solutions, respectively.

(1) The leader chain fork: it is the first important problem to solve, since it is the leader who guarantees the security of Tx blocks. Here, we will employ the corresponding solution in [13]. For ease of reading, we now recap it. Assume that there are k conflicted leader blocks lb_t^i, where $i \in \{0, 1, \ldots, k-1\}$. Each edge node concatenates

these k block header hash strings $H(lb_t^i)$ in a uniform order (e.g., from low to high) as

$$\text{Hash} \leftarrow H\left(H\left(lb_t^1\right) \| H\left(lb_t^2\right) \| \ldots \| H\left(lb_t^k\right)\right). \quad (12)$$

Then, the final winner leader block is

$$i = \text{Hash.substring}(0, k-1) \bmod k \quad (13)$$

(2) The Tx chain fork: assume that N_i is the previous leader node, Node_j the new leader node packed by a new leader block lb_t, t the time stamp of this new leader block. After the generation of this new leader block lb_t, the previous leader node N_i and some other edge nodes might receive lb_t with some delay, due to the bad network. Thus, N_i still keeps on packing and broadcasting some Tx blocks. Assume that these Tx blocks are denoted as $\text{BS}_{N_i} = \{b_{t1}, b_{t2}, \ldots, b_{tm}\}$, simultaneously, the new leader Node_j also is packing and broadcasting some Tx blocks. As a consequence, there will be a time overlap between BS_{N_i} and the Tx blocks by the new leader Node_j. It will cause chaos of the transaction verification. Our solution to this Tx chain fork is to set only the Tx blocks with time stamps no later than t valid and remained in the Tx chain, namely $\left\{b_x | b_x \in \text{BS}_{N_i}, x \le t\right\}$. Otherwise, the Tx blocks packed by the previous leader but with time stamps later than t will be all abandoned.

9. Conclusions

In this paper, we propose an edge-computing-based lightweight blockchain (ECLB) framework for mobile systems. In the ECLB framework, the edge nodes play a minor role. As a consequence, the storage and computation pressure at end mobile devices are greatly relieved. The fragmented ledger is employed as the storage format at end mobile devices. In this way, the end mobile devices not only can obtain information of interest but also do not need to store an entire copy of the ledger. Moreover, we design a two-chain structure of a leader chain and a transaction chain. These two chains grow in parallel. It greatly improves the throughput and confirmation latency. In addition, considering the regulation requirements under the permissioned blockchain setting, we specifically design a pluggable, secure, and efficient transaction regulation protocol. Finally, we give some formal security analysis and performance evaluation. It is demonstrated that our ECLB framework is secure and feasible.

Data Availability

All the experimental data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (grant no. 2020YFB1005500), the National Natural Science Foundation of China (grant numbers 62002139, U1736216, and 61902157), the Natural Science Foundation of Jiangsu Province (grant numbers BK20200886 and BK20200888), and the Project funded by China Postdoctoral Science Foundation (grant numbers 2019M651738 and 2019M661753).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *SSRN Electronic Journal*, 2008.

[2] "ANTMINER S9 Hydro Miner," 2018, https://www.bitmain.com/.

[3] "Blockchain Size," 2020, https://bitinfocharts.com/.

[4] B. Vitalik: Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform," 2013, http://Ethereum.org.

[5] M. A. Javarone and C. S. Wright, "From Bitcoin to Bitcoin cash: a network analysis," *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pp. 77–81, ACM, New York, NY, USA, 2018.

[6] "Litecoin: Open Source P2p Internet Currency," 2011, https://litecoin.org/.

[7] H. Wu, L. Wang, and G. Xue, "Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 589–602, 2020.

[8] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu, and L. Liu, "Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7940–7954, 2020.

[9] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[10] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4Forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 50–57, 2018.

[11] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11008–11021, 2018.

[12] W. Chen, Z. Zhang, Z. Hong et al., "Cooperative and distributed computation offloading for blockchain-empowered industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8433–8446, 2019.

[13] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: towards a scalable public blockchain in fog computing of IoT services computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252–262, 2020.

[14] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: a scalable blockchain protocol," in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation (NSDI)*, pp. 45–59, USENIX Association, Boston, MA, USA, April 2016.

[15] Y. Liu, K. Wang, Y. Lin, and W. Xu, "$\mathsf{LightChain}$: a lightweight blockchain system for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3571–3581, 2019.

[16] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 931–948, ACM, New York, NY, USA, 2018.

[17] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: a secure, scale-out, decentralized ledger via sharding," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 583–598, San Francisco, CA, USA, May 2018.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, Berkeley, CA, USA, 2007.

[19] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman & Hall/CRC, London, UK, 2nd edition, 2014.

[20] H. Deng, Z. Qin, Q. Wu et al., "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3168–3180, 2020.

[21] J. R. Douceur, *"The Sybil Attack," Peer-To-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds., pp. 251–260, Springer, Berlin, Germany, 2002.

[22] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in Bitcoin," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS)*, pp. 906–917, Association for Computing Machinery, New York, NY, USA, 2012.

[23] A. Gervais, G. Karame, K. Wst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 23nd ACM SIGSAC Conference on Computer and Communication Security (CCS)*, ACM, Vienna, Austria, October 2016.

[24] A. De Caro and V. Iovino, "JPBC: Java pairing based cryptography," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pp. 850–855, Kerkyra, Greece, July 2011.

[25] Y. Xu, J. Liu, Y. Shen, J. Liu, X. Jiang, and T. Taleb, "Incentive jamming-based secure routing in decentralized internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 3000–3013, 2021.

[26] C. Qu, M. Tao, and R. Yuan, "A hypergraph-based blockchain model and application in internet of things-enabled smart homes," *Sensors*, vol. 18, no. 9, p. 2784, 2018.

[27] E. Gutierrev, T. P. Monath, A. Alava, D. Uriguen, M. Arzube, and R. W Chamberlain, "Epidemiologic investigations of the 1969 epidemic of Venezuelan encephalitis in Ecuador," *American Journal of Epidemiology*, vol. 102, no. 5, 1975.

[28] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 173–178, ACM, New York, NY, USA, April 2017.

[29] K. Karlsson, "Vegvisir: a partition-tolerant blockchain for the internet-of-things," in *Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1150–1158, Vienna, Austria, July 2018.

[30] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (rbac): features and motivations," in *Proceedings of 11th Annual Computer Security Application Conference*, pp. 241–248, New Orleans, LA, USA, December 1995.

[31] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[32] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[33] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, Aarhus, Denmark, May 2005.

[34] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS)*, pp. 89–98, ACM, Alexandria, VA, USA, October 2006.

[35] W. Ding, Z. Yan, and R. H. Deng, "Privacy-preserving data processing with flexible access control," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 363–376, 2020.

[36] Z. Brakerski, D. Cash, R. Tsabary, and H. Wee, "Targeted homomorphic attribute-based encryption," in *Proceedings of the Theory of Cryptography Conference*, pp. 330–360, Springer, Beijing, China, November 2016.

[37] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "PHOABE: securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT," *Computer Networks*, vol. 133, pp. 141–156, 2018.

[38] E. Matusik, T. P. Gibson, K. Cheng, G. G. Dagher, L. Wang, and S. Yu, "Fluorometric assay for N-acetylprocainamide," *Clinical Chemistry*, vol. 21, no. 13, pp. 1899–1902, 1975.

[39] S. Xiong, Q. Ni, L. Wang, and Q. Wang, "SEM-ACSIT: secure and efficient multiauthority access control for IoT cloud storage," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2914–2927, 2020.

[40] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Generation Computer Systems*, vol. 115, pp. 486–496, 2021.

[41] W. C. Garrison, A. Shull, S. Myers, and A. J. Lee, "On the practicality of cryptographically enforcing dynamic access control policies in the cloud," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 819–838, San Jose, CA, USA, May 2016.

[42] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[43] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 190–199, 2015.

[44] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.

[45] K. He, J. Guo, J. Weng, J. Weng, J. K. Liu, and X. Yi, "Attribute-based hybrid boolean keyword search over outsourced encrypted data," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1207–1217, 2020.

[46] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2016.