Hindawi
Mobile Information Systems
Volume 2022, Article ID 4496486, 21 pages
https://doi.org/10.1155/2022/4496486

Research Article

A Secure and Efficient Access Control Scheme for Shared IoT Devices over Blockchain

Yinjuan Deng,[1] **Shangping Wang**,[1] **Qian Zhang**,[2] **and Jifang Wang**[1]

[1]*School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710054, China*
[2]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710054, China*

Correspondence should be addressed to Shangping Wang; spwang@mail.xaut.edu.cn

Received 30 June 2022; Revised 4 December 2022; Accepted 7 December 2022; Published 31 December 2022

Academic Editor: Carlos T. Calafate

The concept of shared IoT devices has attracted much attention from the industry sector, academia, and financial institutions, providing various benefits, such as saving resources, reducing personal expenses, and providing convenience. Although shared IoT devices facilitate people's lives and work, the information exchange is over wireless networks that may suffer from some security attacks such as unauthorized access to a shared device or some private information of legitimate users being leaked. It makes the secure access control to the shared IoT devices become an intractable issue. In order to guarantee the access right of the legitimate users, to prevent the problems of privacy leakage and unnecessary economic disputes, a secure decentralized access control scheme for shared IoT devices is proposed leveraging the technologies of blockchain and a proposed authentication protocol in this paper. The new lightweight authentication protocol is proposed to perform mutual authentication between the user and the IoT device. To protect the privacy of the user, the instruction data are encrypted by a temporary session key negotiated between the user and the IoT device with the help of blockchain which enables nontamperable transactions and prevents central corruption and single point of failure. In our scheme, blockchain is maintained by the gateway nodes an acts as a distributed database and a smart contract for shared service is deployed on it. The smart contract has three functions in our scheme: (1) achieving the prepayment of users and settlement for the service contributor, (2) participating in a verification step during the key negotiation to prevent some malicious behaviour from users or devices, (3) recording the workload of the gateway. Finally, a comprehensive analysis on the safety and reliability of the entire scheme is carried out; extensive simulation experiments are conducted to reveal the authentication protocol is efficient and the scheme is feasible.

1. Introduction

With the progress of science and technology, shared IoT devices such as shared fitness equipment, shared medical appliances, and shared transportation tools bring great convenience for people because they can provide services for people anytime and anywhere. However, in traditional approaches, when using shared IoT devices, the users need to register on a centralized organization with their personal information and pay the cost for the organization. If the designed system does not consider security well, it may lead to the leakage of users' personal privacy information and a trust crisis in the organization. Hence, it becomes a thorny issue to solve the problem of users accessing the shared devices securely after prepayment, meanwhile protecting the service providers' interests and users' privacy.

The traditional access approaches to shared IoT devices are based on the central organization and some simple password authentication method. The central organization issues passwords or smart cards to users [1, 2], and the users present passwords or smart cards to authenticate themselves when accessing IoT devices. However, the hash function is only used in the password-based and smart card authentication algorithm [3], which makes the security of these methods, not enough.

In addition, the access control based on the centralized organization has the risk of a single point of failure which may cause the system to crash. Moreover, the centralized

organization also may leak the user's privacy information and appear the phenomenon of economic settlement errors [4]. Blockchain is a distributed bookkeeping book [5]. The data is stored on the blockchain after the consensus protocols, so it is fair and equitable. At the same time, the smart contract that is a piece of automatically executed code can be run on the blockchain to form an immutable and transparent record [6]. These records have multiple copies on multiple nodes, which can solve the single point of failure problem effectively. We can combine the currency trading function possessed by the blockchain itself with smart contracts to realize the user's payment to the service provider [7], which can eliminate the user's distrust of the central organization during use.

Relative to password or smart card-based authentication methods, the authentication protocol based on public key cryptography can solve the problem of users' real-time access control to shared IoT devices well, can protect users' privacy, and resist multiple attacks [8]. However, the traditional public key cryptography authentication protocols are mostly based on centralized organization. As far as the protocols themselves are concerned, the following issues need to be considered when using public key authentication protocols in the shared Internet of Things environment. Firstly, shared IoT devices have limited computing power and communication bandwidth [8, 9], so it is necessary to design a public key authentication protocol proper for IoT devices. Secondly, some secret values are required to transmit to the central organization through a secure channel by the users and devices during the authentication process in most of the authentication protocols [10, 11]. However, building numerous secure channels for so many distributed sharing devices is too expensive for service providers to afford. Thirdly, a user can only access a shared IoT device after prepaying in advance. However, it is difficult for the public key cryptography algorithm to connect the prepaid status with the authentication protocol. Moreover, in order to prevent impersonation attacks, it is necessary to make two authentications between the shared devices and the users; the first one is for starting the access, and the second one is for ending the access. As far as we know, the two authentications are independent of each other in the existing schemes. Therefore, we are going to integrate the two authentications to reduce the computing cost and communication cost and conduct economic settlement at the same time.

Based on the above considerations, we intend to use the blockchain, a decentralized technology, to replace the central platform. Moreover, we make the blockchain cooperate with authentication protocol based on public key cryptography to jointly complete the access control of users to shared IoT devices. With the use of blockchain and authentication protocol, the users do not need to provide personal information and can use the shared IoT device anytime and anywhere safely. The approach should protect the economic interests of users and service providers. There is no need for the service providers to spend a lot of money to build secure channels. Also, the information recorded on the blockchain can be used to feed back the device status so that the service provider can adjust the supply strategy in a timely manner.

The main contributions of this paper are as follows:

(1) We propose a secure and efficient access control scheme for shared IoT devices over blockchain by a decentralized approach. The scheme enables a user to access a shared device after getting a negotiation key with the device under the help of blockchain. It can protect users' identity information because of no personal information is needed to be submitted to the central organization. The scheme can prevent a single point of failure, and the tamper-proof transaction and the hash of the negotiation key are recorded on the blockchain.

(2) A new authentication protocol suit for IoT devices is proposed for real-time authentication and key negotiation between a user and a shared device. The key is not only the token for the user to access the shared device but also used to protect the data privacy of the user. Moreover, no secret values are required to be transmitted to the central organization. The smart contract based on the proposed authentication protocol is designed to solve economic problems and some security issues.

(3) Security analyses are carried out on the whole system and some possible security risks. The result shows that our scheme is safe and reliable. The efficiency of the authentication protocol and smart contract are tested, and the results indicate that our scheme is efficient and suitable for the shared IoT environment. The proposed authentication protocol makes the verification burden on a user and an IoT device side less than the comparative literature.

2. Related Work

There are many solutions for access control in IoT environments. In this section, these schemes are similar to ours in the scenario and model assumptions are elaborated. They are divided into two categories: centralized access control and decentralized access control based on blockchain.

2.1. Centralized Solutions. There are various schemes available to control the access authority for IoT scenarios based on public key cryptography [10–15]. Alsahlani and Popa designed an authentication scheme for users to access IoT device data through a gateway in the literature [11]. They proposed a new authentication protocol by which the user is authenticated by the device through three factors, namely, password, identity, and biometric information. Some parameters related to the device and user are stored at the gateway to confirm whether a user has the authority to access the device, and they are determined before a user submits a request. Lots of storage space is required when the number of users and devices is large and flexible access is poor, which is more suitable for centralized scenarios. In the literature

[16], the authors proposed a certificate-based authentication protocol between IoT devices, which can be used for mutual access between two devices and is efficient. However, their scheme is at a risk of collusion attacks in the issuance of certificates. Moreover, they assumed the gateway was honest and trustworthy. In the literature [17], a new authentication protocol based on public key cryptography between vehicles is proposed by the authors; in order to match the actual circumstances, they assumed that the gateway is not completely credible. The authentication of a vehicle and a roadside unit is finished first and then the authentication between the two vehicles. Liu et al., in literature [18], designed a centralized and secure access control scheme for sharing devices. They combined several existing signature algorithms to complete the authentications between users and gateways, as well as gateways and devices subtly, so as to realize users' access to shared devices.

In centralized schemes, the central organization has too much power and work burden; once there is a single point of failure, the entire system will crash.

2.2. Decentralized Access Control Solutions. Some solutions have been proposed by researchers to weaken the power of the centre [19, 20] and to adapt to the distributed characteristic of the IoT environment. However, these schemes only reduce part of the centre's power, and the problem of a single point of failure still exists. Blockchain has emerged as a new type of technology and can solve the problems above [21]. The blockchain system is distributed naturally, which puts forward a new idea for solving security problems in the IoT environment. There are literature works using blockchain technology to solve the secure access control problem in certain IoT scenarios. They are divided into two categories regarding the role of blockchain. One type is that the blockchain is used as a storage place only, and the other type is that the blockchain is involved in the calculation process.

2.2.1. Blockchain for Storage. Rathee et al. [22] use blockchain to store the information of IoT devices to ensure the security of users and devices and maintain transparency among various authorities. Moreover, it can reduce false requests from users, reduce the damage to IoT devices, and reduce the changes in user ratings effectively. In [23], in order to prevent the problem of message leakage during data transmission in the wireless network, based on a private blockchain, the authors proposed an authentication and key agreement protocol for the Internet of vehicles. The blockchain records the pointers of the information needed by vehicles in the authentication process. In their solution, mobility is achieved. Sharma et al. described how to integrate smart contracts into the APP of medical Internet of things in [24]. A specific smart contract is designed for the medical IoT, and the designed system can store the records related to the patients. The authors in [25] proposed an authentication scheme for IoT devices and base stations through digital signatures. The data is semidecrypted by the base station and then transmitted to the blockchain for storage. However, both parties must compute pair-based operations which may not be affordable for IoT devices with less computing power. Cui et al. [26] proposed an authentication method for multiple wireless sensors network. A new blockchain model is set up including a private blockchain to store the information of devices in each area and a public blockchain maintained by the entire network to achieve cross-domain access. However, the authentication must be signed by the cluster head node using a digital signature without considering privacy issues. In order to protect the privacy of the user's identity, Zhang et al. put the user's pseudonyms and public keys on multiple blockchains [27]. To get the desired public key, they subtly designed a method to aggregate multiple pseudonyms to get the real one through querying the full nodes from several blockchains. The fog node must be honest and trustworthy, and the user's information needs to be stored on the blockchain in advance. In literature [28], Rathee et al. proposed hybrid architecture for the medical Internet of things. The scheme gives the detailed use of blockchain to ensure the security and transparency of patient data, file accessibility, and the transportation process.

2.2.2. Blockchain Participates in Calculation. In [29], the authors designed multiple private blockchains for multiple areas divided in the environment of the Internet of vehicles. The vehicle and a blockchain node authenticate each other and generate a shared key to control the access right of the vehicle. No identity verification is performed when trust authority distributes keys to vehicles, which are prone to suffer man-in-the-middle attacks. By using blockchain, Vishwakarma and Das [30] established a secure communication channel between devices. Moreover, the cluster head performs the verification of the cluster members on the blockchain. This solution is not suitable for mobile devices. In [31], when a user wants to access a device, he or she needs to register on the blockchain on the Ethereum smart contract and then be authenticated by smart contracts. For access control, the smart contract contains the mapping of all registered IoT devices to their licensed users. The scheme needs to determine whether a user can use the device in advance, which is not suitable for our scenario. Hammi et al. [32] divided the devices into several different areas called bubbles, and each step in the authentication process is recorded as a transaction on the blockchain. Moreover, the authentication only happened between two devices in the same bubble. In [33], Li et al. proposed a multidomain authentication scheme for IoT based on cross-chain technology, which realizes the authentication process by the smart contract. In Almadhoun et al. 's [34] scheme, the access authority is controlled by a smart contract by distributing access tokens for users. However, the verification between the fog node and a user is just by a simple digital signature, without considering the issue of identity privacy. In order to avoid the key escrow problem found in identity-based systems, Mwitende et al. proposed a pairing-based certificateless authenticated key agreement protocol used for the controller (of the device) and a node of the blockchain. A blockchain-based architecture was proposed for the protocol in reference [35]. However, we found it requires more complicated calculations.

After discussing the related literature, based on various security considerations, different authors have proposed a variety of authentication protocols in the related literature to prove the legitimacy when accessing each other or communicating with each other in various IoT environments, such as medical environment, industrial IoT environment, and Internet of vehicles in centralized schemes. In the decentralized scheme, the authors have designed different system architectures for different blockchain-based scenarios. The main contributions including new architectures are proposed combined with the cryptographic protocol, security databases or transmission channels are established through the blockchain, and access rights are automatically controlled by the designed smart contracts.

Thanks to the ideas and related knowledge, we find that the literature for access control in the IoT environment have the following shortcomings for our scenario:

(1) The scheme based on access token or access list need to define the access authority between a user and a device with a certain mapping in advance, while we need real-time access control meanwhile realize instant payment without a central institution. In some decentralized schemes, too much information is stored on the blockchain, and smart contract undertakes too many operations.

(2) Few of the schemes based on public key cryptography consider the mobility of the devices or need heavy calculations. Some schemes only realize mutual authentication and do not negotiate keys to establish a secure channel. Most schemes require a secure channel in the process of access control, which is difficult to achieve in the distributed Internet of things environment. Furthermore, most of them assume the gateway is safe and reliable, which is only suitable for the communication model in their scenario.

However, there is no access control scheme for shared IoT devices under the decentralized platform. Considering the use environment and characteristics of shared IoT devices, we propose a new decentralized architecture based on blockchain in this paper and combine the proposed public key authentication protocol to perform the access control effectively. A new smart contract is deployed on the Ethereum blockchain according to our authentication protocol. However, our scheme keeps the information secure and finishes payment issues when users use the shared devices. Especially under the assumption that the gateway is not trusted, considering the mobility of the device and protecting the user's privacy, the computation efficiency of a user and a device is improved by 50% compared with the scheme that has the same hypothesis.

3. Preliminaries

3.1. Mathematical Problems Used in Cryptographic Algorithms

3.1.1. Elliptic Curve. Suppose p is a large prime number, $GF(p)$ represents a finite field, and all points on the curve $y^2 = x^3 + ax + b \bmod (p)$ and an infinite point form an elliptic curve $E_p(a,b)$. If $a,b \in GF(p)$, $4a^3 + 27b^2 \neq 0 \bmod (p)$, and we call $E_p(a,b)$ as a nonsingular elliptic curve [36]. All points on the curve $E_p(a,b)$ form an additive cyclic group, where the addition operation is $G + G + ... + G = kG$ (add k times), for $k \in Z_l$, and l is the order of the group, G is a generator of the group, and we also call kG as a scalar multiplication.

3.1.2. Difficulty Assumption. The discrete logarithm problem on the elliptic curve: suppose $Q = nG$ is known as a random point on an elliptic curve $E_p(a,b)$ and G is a generator of the curve; then, it is difficult to calculate n.

3.2. Blockchain and Smart Contract. Blockchain is a kind of chain database, which contains transaction blocks, and each block contains multiple transactions. These transactions can be ordinary currency transactions or data exchange records [37]. The blockchain has the characteristic of decentralization, and the data on it is immutable. Transactions on the blockchain are public and stored in a distributed manner, and anyone can inquire them at anytime and from anywhere. Users with blockchain addresses can post transactions at any time. With the development of blockchain architecture and technology, smart contracts can be deployed on blockchain such as Ethereum and Hyperledger. It is a computer program that can execute contract terms automatically [38]. A smart contract is a digital protocol between communication parties based on predefined rules, without the need of a trusted third party. As long as the conditions are met, it can be executed automatically without human interference. The smart contract is executed by external calls and functions. Through these calls and functions, the smart code is executed and events are generated. These events are broadcasted to all the participants, and finally, the smart contract and the transaction will be packaged into the block [39].

4. System Model and Assumptions

There are four entities and one virtual component in our system. The four entities are a certificate authority (CA), users, gateways, and the shared IoT devices. Blockchain is a virtual component, such as the Bitcoin blockchain or the Ethereum blockchain maintained by the gateway nodes.

CA is a trust centre and is responsible for issuing certificates for IoT devices and gateways in the initialization phase, and those certificates are used for identity verification later.

A user uses a device and transmits instructions to the device. There may be malicious gateways and devices, so the legitimacy should be authenticated before the user receives messages from them. Moreover, he cares about the privacy of his identity and the confidentiality of his instructions to prevent malicious people from obtaining information such as lifestyle habits, movement trajectories, and health status by analysing instructions. Hence, the service instruction is sent to the device in the form of ciphertext encrypted by a session key negotiated with the device.

The gateway nodes (GWNs) constitute a network layer and are intermediaries between users and devices. They are run by some communication operators and get a certain amount of remuneration by providing computing power for devices and users. In the shared IoT device environment, some idle gateways that meet the conditions can also join this network layer and earn a certain amount of profit. The gateways are untrustworthy, and there may be fake or unauthorized gateways, so they must be verified.

Shared IoT devices have weak communication and computing capabilities [9]. They perform the corresponding operation after receiving the instruction from users. A shared device is untrustworthy, it may be attacked, or a counterfeit device pretends to be a legitimate device for illegal economic benefit.

Blockchain (short for BC) is an immutable database maintained by gateway nodes. The identity and status information of the legitimate devices are stored on the blockchain. Also, some operations that are agreed in advance and prevent being modified are deployed on the blockchain in the form of smart contract. The smart contract used to pay for the entities and verifies the key information in the key negotiation process, which can ensure the reliability and security.

The communication and connection between various components are shown in Figure 1(a), the system architecture of the proposed scheme. Figure 1(b) gives the brief process of the proposed scheme to show the rough flow when a user accesses a shared IoT device. The solid line represents offline communication. The dotted line represents wireless communication over a public channel.

4.1. Assumptions

(1) Assume that a user and the device use the same gateway when sending an access request and ending an access to a shared device

(2) Assume that the ultimate aim of gateways is for economic benefits; the gateways with successful authentication follow the access control process but are interested in the data and privacy of users

4.2. Attack Model

(1) The attacker intercepts the information on the public channel, attempts to obtain the user's identity information or instruction information, and attempts to act as an intermediary to negotiate the key with the shared device.

(2) The attacker may pretend to be real users to use a shared device, and an illegal gateway may pretend to be the legal gateway and obtain illegal service fees.

(3) A malicious user and a gateway may launch a collusion attack and not give the prepayment. Moreover, the gateway and a device may conspire to defraud the service fee.

5. Design Goals

Considering the requirements of the system architecture and security of the authentication protocol, the following points are needed.

5.1. Decentralization. There is no need for users to interact with the certificate authority or other central institutions. The system can still run steadily even if the data on a central server is damaged or lost. The device providers or any organization cannot change the data at will, and if there is a dispute, it can be traced.

5.2. Mutual Authentication and Key Negotiation. In order to achieve a legitimate user (the user who has prepaid for an item) to access the selected shared device, mutual authentications between the user and the nearest gateway and the gateway and the device should be carried out to ensure the legitimacy of the communication objects. A key negotiation between the user and the device should be completed, which is used for encrypting the instructions sent from the user to the device so as to protect the user's information privacy.

5.3. The Anonymity of Users. In order to protect users' personal identity information and prevent them from being leaked on public platforms, such as phone numbers, the user's identity information should be kept secret. Pseudonyms are used for authentication and key negotiation. Any adversaries cannot infer the real identity of the user from the messages sent by the user.

5.4. Access List Is Not Required. It is random for a user to access a shared device; an access permission mapping between users and devices is impossible, leading to the access list unable to be used for controlling the user's access authority.

5.5. Central Server Is Not Needed to Be Online Always. There is no need for a central server online all the time for the actual availability of the system.

5.6. Resisting Common Attacks. Denial of service attacks can be initiated by malicious users, role replacement attack, collusion attack, man-in-the-middle attack in the key negotiation process, and replay attack.

6. The Proposed Scheme

In the traditional approaches, a user needs to purchase keys from the central organization to access a shared IoT device [18]. In order to verify his validity, a message signed using one of the keys is sent to the gateway and then mutual identity authentication and key negotiation are performed between the user and the gateway. Finally, a key k_1 is negotiated successfully. Similarly, an identity-based

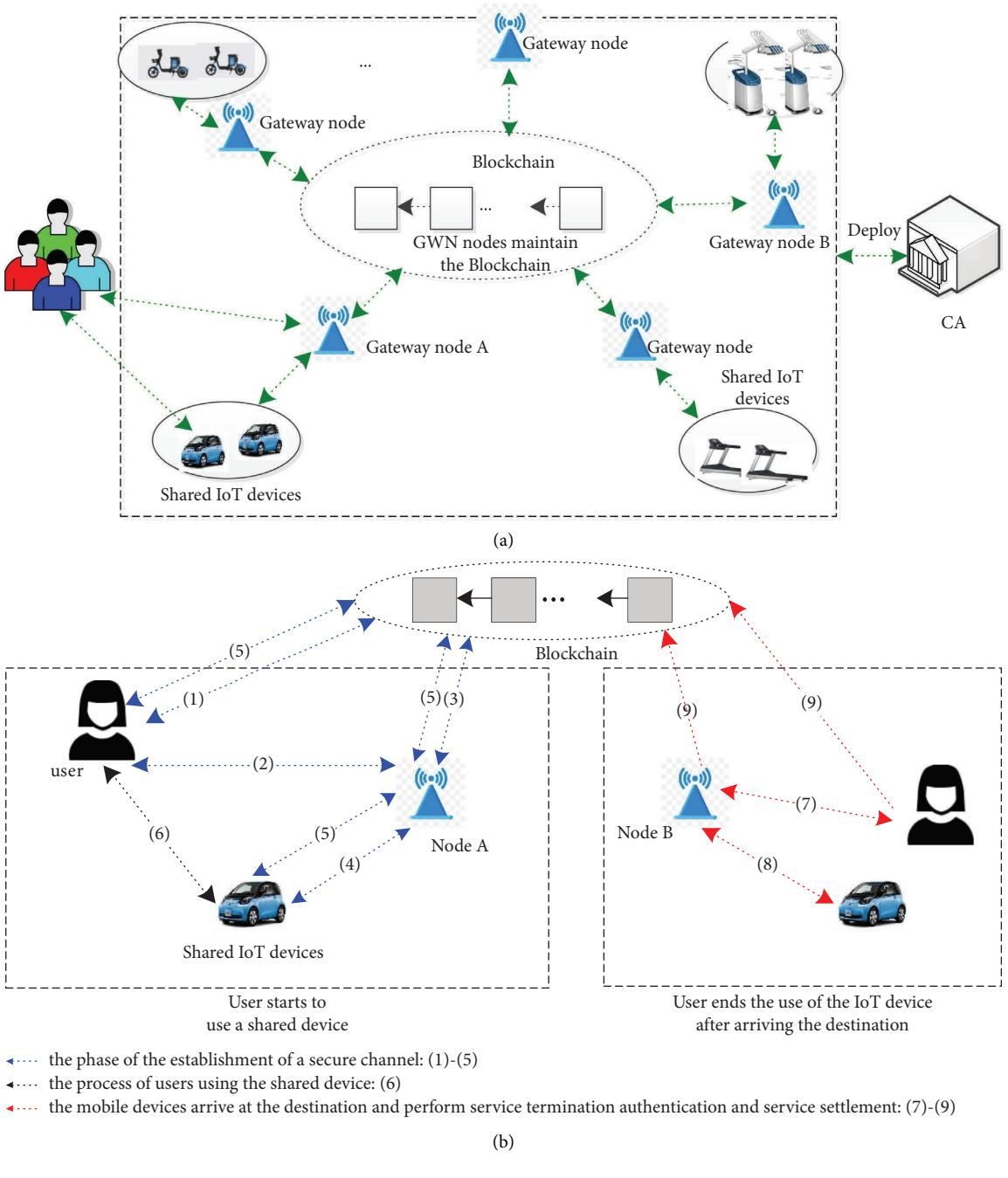

Figure 1: (a) System architecture of the proposed scheme. Note: This figure contains all the components in the scheme and their interactions. The green line with arrows indicates wireless communication. The green solid line indicates offline communication. The CA deploys and initializes the whole system. The user communicates with the shared devices through the gateways before being granted access and can communicate with the device directly after being authenticated. Because of limited communication capacity, the shared devices upload information to the blockchain through the gateway. (b) The access process of a user to a shared IoT device. Note: (1) When a user wants to access a shared IoT device, he/she searches for a special service item on the blockchain and makes the prepayment. (2) The user and nearest gateway node A authenticate each other using the proposed authentication protocol. (3) The gateway node checks whether the user has prepaid for the service item on the blockchain. (4) The gateway and the shared IoT device authenticate each other using the proposed authentication protocol. (5) The user transmits parameters to the smart contract, and the device transmits parameters through the gateway to complete the final step of the authentication. (6) The user sends a specific instruction in the form of ciphertext to the shared device, and the device decrypts it with the negotiation key. Then, the service starts. For example, in the scene of a shared car, it goes from place A to place B. When the user and shared device such as the shared car arrive at the destination near a new gateway node B, the user wants to end the use of the shared car. In order to prevent impersonation attacks and clearing the charge, they begin to authenticate each other. (7) The user and the new gateway do the mutual authentication. (8) The device authenticates the new gateway. (9) The user, the new gateway, and the device pass relevant parameters to the smart contract for settlement, and the service is ended.

authentication method is used for the gateway and the device to authenticate each other and negotiate another key k_2. When the service is starting, the user encrypts the instruction by k_1 and sends the ciphertext c_1 to the gateway and then the gateway decrypts it and re-encrypts the instruction using k_2 and sends the new ciphertext c_2 to the device; after decrypting the new ciphertext c_2, the device can get the instruction. Lastly, in order to settle the service fee for the gateway, an aggregated signature is used to calculate the times of service provided by the gateway and the information is sent to the central organization for settlement. Some scholars have studied authentication protocols suitable for IoT environments for access control [10, 17], but the process of paid access is not involved.

In this section, based on the blockchain and a proposed certificate-based authentication protocol, a specific decentralized access control scheme for the shared IoT device environment is described in detail. A user can access a device safely without disclosing their real identity. The authentications and service settlement are finished by the proposed authentication protocol collaborating with the smart contract. In order to resist a single point of failure, central corruption and other security issues proposed in Section 5, and to improve system efficiency, a new decentralized architecture is proposed, in which the blockchain records some authentication information and the smart contract replaces some complex cryptographic algorithms to record the workload of gateways and settle the problem of fair payment automatically.

The process in the proposed scheme includes the following steps:

(1) *System Initialization*. Selecting the curve equation and parameters required in the system and deploying the smart contract on the blockchain are included in system initialization.

(2) *Registration*. IoT devices and GWN register with the CA, and the CA issues certificates for them. At the same time, the CA uploads the public information of legal IoT devices, such as service lists and charges, device status, and identity identifier, to the blockchain for storage.

(3) *Establishment of a Secure Channel*. After the prepayment is successful, the user builds a secure communication channel with a device through the gateway and blockchain.

(4) *Service Process*. The user sends the encrypted service instructions to the device to perform the service process.

(5) *End of Service*. The user requests to end the service. The settlement is completed with the cooperation of the gateway and the smart contract.

Table 1 shows the specific symbols used in our scheme.

6.1. System Initialization. The process is done by the CA and service provider (for the sake of brevity, we omit the service provider in the system architecture) jointly, some appropriate parameters are selected, and a smart contract is deployed on the blockchain. This process is offline and done in advance.

The CA chooses a nonsingular elliptic curve $E_p(a, b)$ and selects a generator G of order n on the curve and an anticollision hash function h: $\{0,1\}^* \longrightarrow Z_p^*$. Then, the CA chooses a private key $k_{CA} \in Z_p^*$ for himself and keeps it secretly, calculates its public key $K_{CA} = k_{CA} \cdot G$, gets public-private key pairs (K_{CA}, k_{CA}), and publishes the system public parameters $\{E_p(a,b), G, h(\cdot), K_{CA}, n\}$. At the same time, CA deploys the shared service contract and gateway certificate verification contract on the blockchain, which is used for prepayment, recording the workload of the gateway, verifying the negotiated key, and performing the settlement of the service.

6.2. Entity Registration. Device registration and gateway registration are included. In order to authenticate and negotiate a key, the two entities must apply for certificates at the certificate authority (CA). The certificate of each device and gateway is unique and long term. Registration is only performed once in the entire process and is done in advance.

6.2.1. Registration of an IoT Device

RD.1: An IoT device Dev_j selects a random number $k_{\mathrm{Dev}_j} \in Z_p^*$ as its own private key and calculates its public key $K_{\mathrm{Dev}_j} = k_{\mathrm{Dev}_j} \cdot G$. The device applies for a blockchain address $B\mathrm{add}_{\mathrm{Dev}_j}$.

RD.2: The IoT device submits its registration request information $RDe = \left\{ID_{\mathrm{Dev}_j}, K_{\mathrm{Dev}_j}, \mathrm{Profile}_{\mathrm{Dev}_j}\right\}$ to the CA in the offline model. ID_{Dev_j} is its identity identifier, $\mathrm{Profile}_{\mathrm{Dev}_j} = \left\{B\mathrm{add}_{\mathrm{Dev}_j}, \mathrm{Location}, \mathrm{SerList}, \mathrm{Status}\right\}$, Location is the location of the device, SerList includes the service names, the function of the device, and their charges, Status is the status of the device, 0 is damage and not available, 1 indicates that the device can be used normally, and 2 means the device is in use.

RD.3: After getting *RDe*, CA issues certificate $\mathrm{Cert}_{\mathrm{Dev}_j} = \left\{\mathrm{ECDSA}_{k_{CA}}(h(K_{\mathrm{Dev}_j} \| ID_{\mathrm{Dev}_j})), ID_{\mathrm{Dev}_j}, \mathrm{deadline}\right\}$ for the IoT device and stores the public information of the device $\mathrm{identify} = \left\{K_{\mathrm{Dev}_j}, B\mathrm{add}_{\mathrm{Dev}_j}, \mathrm{Cert}_{\mathrm{Dev}_j}\right\}$ and $\mathrm{Profile}_{\mathrm{Dev}_j} = \{\mathrm{Location}, \mathrm{SerList}, \mathrm{Status}\}$ on the blockchain.

6.2.2. Registration of a Gateway

RG.1: A gateway GWN_k selects a random number $k_{\mathrm{GWN}_k} \in Z_p^*$ as its own private key and calculates its public key $K_{\mathrm{GWN}_k} = k_{\mathrm{GWN}_k} \cdot G$. The device applies for a blockchain address $B\mathrm{add}_{\mathrm{GWN}_k}$.

RG.2: The gateway gives its own registration information $\mathrm{RGWN} = \left\{ID_{\mathrm{GWN}_k}, K_{\mathrm{GWN}_k}, B\mathrm{add}_{\mathrm{GWN}_k}\right\}$ to the CA in the offline model.

Table 1: Symbol description.

Symbols	Description
CA	Certificate authority
u_i	The ith user
GWN_k	The kth gateway
Dev_j	The jth shared device
SK	Negotiated session key
S_i	The ith service item
$E_{SK}(\cdot)$	The symmetric encryption algorithm with the session key SK
$\mathrm{Cert}_{u_i}, \mathrm{Cert}_{\mathrm{GWN}_k}, \mathrm{Cert}_{\mathrm{Dev}_j}$	The certificate of ith user, kth gateway, and jth device
$B\mathrm{add}_{u_i}, B\mathrm{add}_{GWN_k}, B\mathrm{add}_{\mathrm{Dev}_j}$	The blockchain address of ith user, kth gateway, and jth device
$k_{u_i}, k_{\mathrm{GWN}_k}, k_{\mathrm{Dev}_j}$	The secret key of ith user, kth gateway, and jth device
$K_{u_i}, K_{\mathrm{GWN}_k}, K_{\mathrm{Dev}_j}$	The public key of ith user, kth gateway, and jth device
$\mathrm{ECDSA}_{K_{CA}}(\cdot)$	Elliptic curve signature algorithm with the secret key of CA

RG.3: After getting the registration information of the gateway, CA issues a public certificate $\mathrm{Cert}_{\mathrm{Dev}_j}$ including signature $\mathrm{ECDSA}_{k_{CA}}(K_{\mathrm{GWN}_k} \| B\mathrm{add}_{\mathrm{GWN}_k} \| ID_{\mathrm{GWN}_k})$, identity ID_{Dev_j}, and deadline to the gateway GWN_k.

After getting the certificate, the gateway calls the gateway certificate verification process to verify the legitimacy of the certificate, and if the verification passes, the contract address is returned to the gateway.

The CA stores public information of the gateway on blockchain as $\{K_{\mathrm{GWN}_k}, B\mathrm{add}_{\mathrm{GWN}_k}, \mathrm{Cert}_{\mathrm{GWN}_k}\}$.

The above studies can be regarded as the preparation work done by the CA. To avoid fake gateways and devices, both the gateway and device need to be authenticated when a user wants to use a shared device. At the same time, considering the user's privacy and data confidentiality, the user needs to negotiate a session key with the shared device in real time. The validity period of the key is from the beginning to the end of the use.

6.3. Establishment of a Secure Channel. When a user wants to use a shared IoT device, he can search the blockchain to find a proper device and service item. Then, he starts to establish a secure channel with the shared IoT device. To guarantee the reliability of the system, the smart contract participates in the last step of key negotiation, which is nontamper and traceable. Moreover, it can also be a proof of the success of key negotiation, as well as settle the payment for the IoT device and the gateway after the service is ended.

6.3.1. AU.1.1: $u_i \longrightarrow BC$. A user searches for a service on the blockchain and makes prepayment.

A user queries the service list on the blockchain, selects a device Dev_j nearby and a service S_i provided by the device, and gets ide ntify $= \{K_{\mathrm{Dev}_j}, B\mathrm{add}_{\mathrm{Dev}_j}, \mathrm{Cert}_{\mathrm{Dev}_j}\}$ of the device. Prepayment is done on the smart contract by the user, and if the amount of prepayment meets the payment requirements, a confirmation message will be returned to the user by the smart contract.

6.3.2. AU.1.2: $u_i \longrightarrow \mathrm{GWN}_k$: msg_{U1}. The user computes his authentication information and sends it to a gateway.

After receiving the confirmation message, in order to exclude invalid gateways, a mutual authentication with the nearest gateway GWN_k will be done by the user, as well as negotiating a key with the shared device in real time. The user calculates $\sigma_{u_i} = l_i^{-1}(k_{u_i} \cdot h(K_{u_i} \| ID_{\mathrm{Dev}_j} \| S_i \| R_{u_i} \| T_{i1}) + h(r_i \| k_{u_i} \| B\mathrm{add}_{u_i} \| T_{i1}))\mathrm{mod}\ p$, where $l_i, r_i \in Z_p^*$ are selected randomly and kept secretly by the user, $R_{u_i} = h(r_i \| k_{u_i} \| B\mathrm{add}_{u_i} \| T_{i1}) \cdot G$ is public, and T_{i1} is the time the message is sent. The signature $\mathrm{Sig}_{u_i} = (\sigma_{u_i}, L_i)$ is used for generating a shared key and authenticating himself by the gateway, where $L_i = l_i \cdot G$ is public.

Then, the user sends his authentication information $\mathrm{msg}_{U1} = \{\mathrm{Sig}_{u_i}, K_{u_i}, B\mathrm{add}_{u_i}, ID_{\mathrm{Dev}_j}, S_i, R_{u_i}, T_{i1}\}$ and the message of the successful prepayment to the nearest gate way GWN_k, where T_{i1} is the time the message is sent by the user.

6.3.3. AG.2.1: The Gateway Authenticates the User. After receiving the message msg_{U1} from the user, the gateway queries the prepayment status on the blockchain. If it is successful, the authentication will continue as follows:

(1) To prevent replay attacks, the freshness of the message is verified firstly; if $|T_{\mathrm{cur}} - T_{i1}| < \Delta t$, then it performs step (2), else stops. T_{cur} is the time of receiving the message by the gateway, and Δt is the time delay permitted.

(2) It verifies the validity of the user's identity. The gateway calculates these equations $h_1 = h(K_{u_i} \| ID_{\mathrm{Dev}_j} \| S_i \| R_{u_i} \| T_{i1})$, $v_1 = {\sigma_{u_i}}^{-1} \cdot h_1 \cdot K_{u_i} \mathrm{mod}\ p$, and $v_2 = {\sigma_{u_i}}^{-1} \cdot R_{u_i} \mathrm{mod}\ p$. If equation $v_1 + v_2 = L_i$ holds, the verification is passed; otherwise, it is rejected.

6.3.4. AG.2.2: $\mathrm{GWN}_k \longrightarrow \mathrm{Dev}_j$: $\{\mathrm{msg}_{G1}, \mathrm{msg}_{U1}'\}$ *and* $\mathrm{GWN}_k \longrightarrow u_i$: $\{\mathrm{msg}_{G1}\}$. The gateway calculates its own authentication information and sends it and the part of the user's information to the device and, at the same time, sends its authentication information to the user.

If the user is authenticated successfully, the gateway GWN_k calculates its signature $\sigma_{\mathrm{GWN}_k} = l_k^{-1} \cdot k_{\mathrm{GWN}_k} \cdot h(K_{\mathrm{GWN}_k} \| B\mathrm{add}_{\mathrm{GWN}_k} \| ID_{\mathrm{GWN}_k} \| \mathrm{Cert}_{\mathrm{GWN}_k} \| T_{k1}) \bmod p$, where $l_k \in Z_p^*$ is a secret number selected randomly by the gateway. Then, the gateway sends its authentication message $msg_{G1} = \{\mathrm{Sig}_{\mathrm{GWN}_k}, ID_{\mathrm{GWN}_k}, T_{k1}\}$ to the device, and the user's message $msg_{U1}' = \{\sigma_{u_i}, K_{u_i}, Badd_{u_i}, S_i, R_{u_i}, T_{i1}\}$ is sent to the device Dev_j to calculate the shared secret key with the gateway; the signature is $\mathrm{Sig}_{\mathrm{GWN}_k} = (\sigma_{\mathrm{GWN}_k}, L_k)$, $L_k = l_k \cdot G$ is public, and T_{k1} is the time the message is sent by the gateway. To save the time, the L_k can be precalculated. And the gateway sends its own authentication message $\mathrm{msg}_{G1} = \{\mathrm{Sig}_{\mathrm{GWN}_k}, ID_{\mathrm{GWN}_k}, T_{k1}\}$ to the user to prove its legitimacy immediately.

6.3.5. AD.3.1: The Authentication Process of the Gateway. The authentication process of the gateway is similar to the user and the device, as follows.

(1) The user checks the validity of the certificate of the gateway on the blockchain. If it is valid, he gets the public key K_{GWN_k} of the gateway, blockchain address $B\mathrm{add}_{\mathrm{GWN}_k}$, and certificate $\mathrm{Cert}_{\mathrm{GWN}_k}$ from the blockchain. (2) In order to verify the authenticity of the gateway, he checks whether the following equation holds:

$$\sigma_{\mathrm{GWN}_k}^{-1} \cdot h \cdot PK_{\mathrm{GWN}_k} = L_k, \tag{1}$$

where $h = h(K_{\mathrm{GWN}_k} \| B\mathrm{add}_{\mathrm{GWN}_k} \| \mathrm{Cert}_{\mathrm{GWN}_k} \| T_{k1})$.

If the certificate is valid and (1) holds, a successful authentication message is sent to the gateway by the user; otherwise, it returns a failure. And, if it is authenticated successfully by the device, then the device does the following steps; else, it returns a failure.

6.3.6. AD.3.2: $\mathrm{Dev}_j \longrightarrow \mathrm{GWN}_k$: $\{\mathrm{msg}_{D1}\}$, $\mathrm{Dev}_j \longrightarrow BC$: SKV. The device sends its own authentication information to the gateway and passes the hash of the negotiated key calculated by itself as a parameter to the smart contract.

If the authentication of the gateway passes, the device regards the gateway as credible, then calculates $msg_{D1} = \{\mathrm{Sig}_{\mathrm{Dev}_j}, ID_{\mathrm{Dev}_j}, R_{\mathrm{Dev}_j}, T_{j1}\}$ as its authentication information, and sends it to the gateway, where $\mathrm{Sig}_{\mathrm{Dev}_j} = (\sigma_{\mathrm{Dev}_j}, L_j)$ and $\sigma_{\mathrm{Dev}_j} = l_j^{-1} \cdot (k_{\mathrm{Dev}_j} \cdot h(\mathrm{Cert}_{\mathrm{Dev}_j} \| R_{\mathrm{Dev}_j} \| T_{j1}) + h(r_j \| k_{\mathrm{Dev}_j} \| T_{j1})) \bmod p$, $l_j \in Z_p^*$ is selected arbitrarily and kept secretly by the device, and $R_{\mathrm{Dev}_j} = h(r_j \| k_{\mathrm{Dev}_j} \| T_{j1}) \cdot G$ and $L_j = l_j \cdot G$ are public. And, we remark $h(\mathrm{Cert}_{\mathrm{Dev}_j} \| R_{\mathrm{Dev}_j} \| T_{j1}) = h_3$. To save the time, L_k can be precalculated. Otherwise, it terminates the subsequent process. Then, the device calculates the key shared with the user using R_{u_i} and $B\mathrm{add}_{u_i}$ as

$$SK = h\left(h\left(r_j \left\| k_{\mathrm{Dev}_j} \right\| T_{j1}\right) \cdot R_{u_i} \left\| \mathrm{Cert}_{\mathrm{Dev}_j} \right\| B\mathrm{add}_{u_i}\right). \tag{2}$$

And, it calculates the verifier $\mathrm{SKV} = h(SK \| T_{j1} \| B\mathrm{add}_{\mathrm{GWN}_k})$ of the key and sends SKV to the smart contract to verify later.

6.3.7. AG.4: $\mathrm{GWN}_k \longrightarrow u_i$: $\{\mathrm{msg}_{D1}'\}$. The gateway verifies the legitimacy of the device and forwards the negotiation key information of the device used to calculate the negotiation key to the user.

After receiving msg_{D1}, the gateway queries the information $\{K_{\mathrm{Dev}_j}, B\mathrm{add}_{\mathrm{Dev}_j}, \mathrm{Cert}_{\mathrm{Dev}_j}\}$ of the device on the blockchain and uses the method similar for AG.2.1 to verify the device. If the verification is passed, the device is considered as legal and the gateway sends $\mathrm{msg}_{D1}' = \{\sigma_{\mathrm{Dev}_j}, ID_{\mathrm{Dev}_j}, R_{\mathrm{Dev}_j}, T_{j1}\}$ to the user; otherwise, it is terminated.

6.3.8. AU.5: $u_i \longrightarrow BC$: $\{\mathrm{SKV}, B\mathrm{add}_{\mathrm{GWN}_k}\}$. The user passes the hash of negotiation key calculated by himself as a parameter to the smart contract.

The user utilizes the public information of the device $\mathrm{msg}_{D1}' = \{\sigma_{\mathrm{Dev}_j}, ID_{\mathrm{Dev}_j}, R_{\mathrm{Dev}_j}, T_{j1}\}$ and $\mathrm{identify} = \{B\mathrm{add}_{\mathrm{Dev}_j}, K_{\mathrm{Dev}_j}, \mathrm{Cert}_{\mathrm{Dev}_j}\}$ stored by the user temporarily to calculate the shared key with the device as

$$SK' = h\left(h\left(r_i \| k_{u_i} \left\| B\mathrm{add}_{u_i} \right\| T_{j1}\right) \cdot R_{\mathrm{Dev}_j} \left\| \mathrm{Cert}_{\mathrm{Dev}_j} \right\| B\mathrm{add}_{u_i}\right). \tag{3}$$

Furthermore, he calculates the equation $\mathrm{SKV}' = h(SK' \| T_{j1} \| B\mathrm{add}_{GWN_k})$ and then sends SKV′ and the blockchain address of the gateway $B\mathrm{add}_{\mathrm{GWN}_k}$ to the smart contract for verification.

6.3.9. ASM.6. The smart contract receives the data submitted by the user and verifies whether the equation SKV = SKV′ holds. If the equation holds, the smart contract returns a message indicating that the secret key is negotiated successfully to the user and the device. Moreover, it sends a certain amount of labour fees to the blockchain address of the gateway and, at the same time, starts to time the user using the device. The specific process is shown in Figure 2 in the form of a flow chart when the smart contract is going to confirm whether the authentication between the user and device is successful or not.

The secure channel establishment phase is the most important stage of this paper. In order for readers to understand this phase clearly, we summarized and sorted out the contents of Section 6.3 and showed the messages sent and operations performed by each component in Figure 3.

6.4. Service Process. The user u_i sends execution command $\mathrm{msg}_{S1} = E_{SK}(\mathrm{cmd})$ to the shared device Dev_j, for $\mathrm{cmd} \in S_i$ is a specific instruction from the user. For example, in

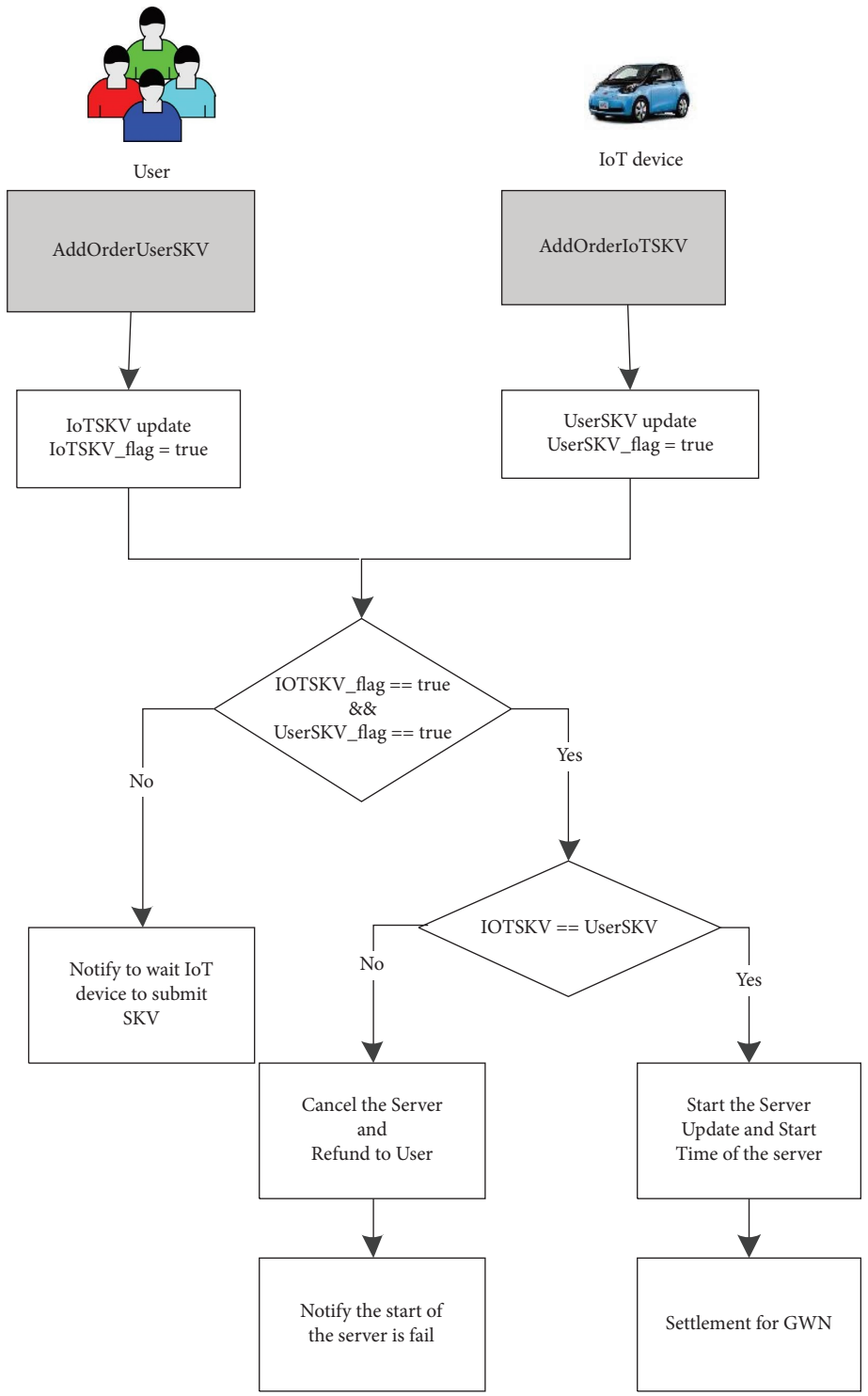

Figure 2: Secret key negotiation confirmation process in the smart contract. Note: (I) The user calls the interface AddOrderUserSKV() of the user to verify the negotiation key, updates the authentication message of the user's negotiation key UserSKV and sets the authentication message update flag of user UserSKV_flag as true. (II) The IoT device calls the interface AddOrderIoTSKV() of the IoT device to verify the negotiation key, updates the authentication message of the user's negotiation IoTSKV, and sets the authentication message update flag of the IoT device IoTSKV_flag as true. (III) When the user or IoT device calls the interface to verify the negotiation key, they can judge whether both parties have finished the update of SKV by verifying the message update flag UserSKV_flag and IoTSKV_flag. If the update is not finished, it notifies the user or IoT device to wait for another to update the SKV within a certain period of time. (IV) When the SKV of both the user and IoT device has been updated, it judges whether UserSKV is equal to IoTSKV; if the SKVs of both parties are not equal, it means that the secret key negotiation between the parties fails, the fee will be refunded to the user, and both parties will be notified of the failure of this service. (V) If the SKVs of both parties are equal, the service starts and start time is recorded. Furthermore, it pays for GWN to provide services for the secret key negotiation.

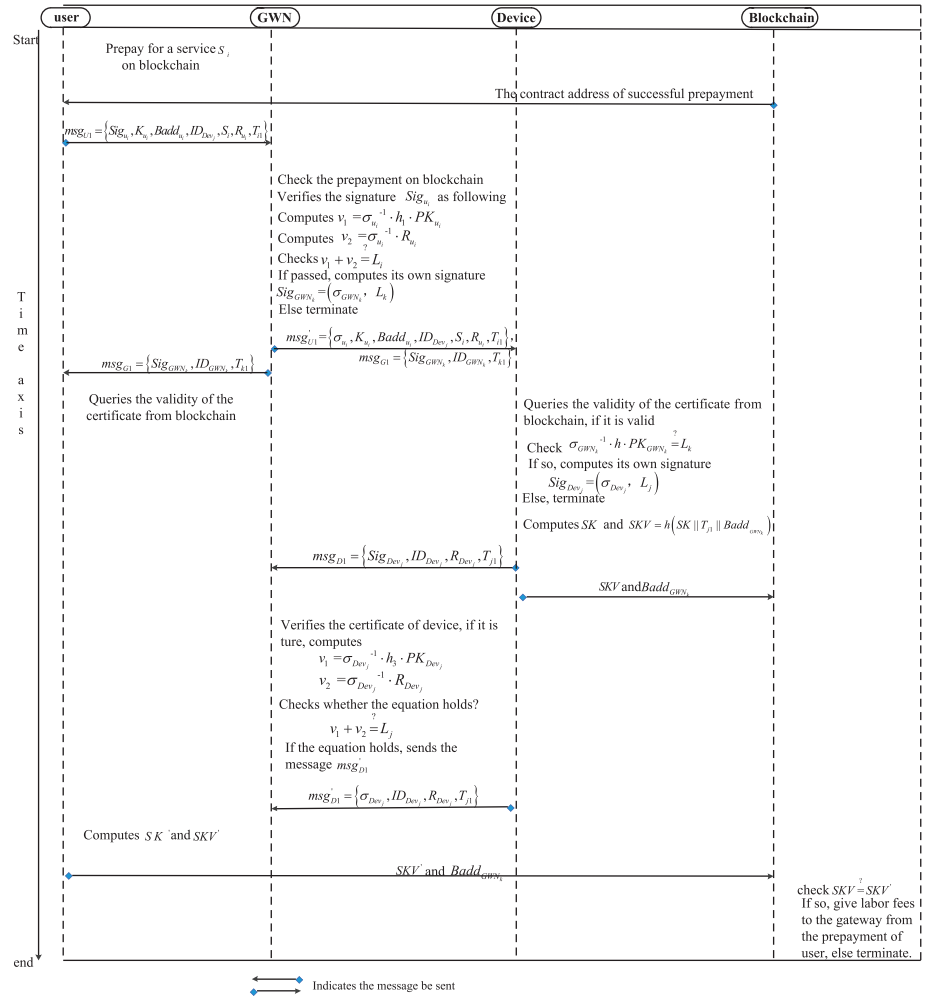

FIGURE 3: A brief summary of the secure channel establishment process for a user and a device. Note: The upper content is carried out first, and the lower content is carried out later. The content between two dotted lines indicates local specific calculations or operations. For example, the sentence "Computes SK' and SKV'" at the bottom left of Figure 3 represents the user's need to compute SK' and SKV'.

driverless driving, "open the door and go somewhere." The device Dev_j decrypts msg_{S1} with the negotiated key, executes the command, and returns a successful execution message.

6.5. End of the Service. There are two situations when the user applies for ending the service. One is that the shared device is still and the gateway is still the original one, such as a treadmill. Another is that the device is mobile, such as a shared car and wearable medical device and a new gateway is required for communication.

6.5.1. End Service of the Stationary Device. The original gateway GWN_k is still in use. The user sends the end instruction to the gateway and the device. The gateway calls static device settlement function in smart contract and sends blockchain address $B\text{add}_{u_i}$ of the user, its own blockchain address $B\text{add}_{\text{GWN}_k}$, and the device's blockchain address $B\text{add}_{Dev_j}$ to the contract to settle the service cost. Then, the user's prepayment is transferred to the accounts of the gateway and device according to a certain rule (the regulation is laid down when the smart contract is deployed) which may be related to the time of use and the unit use price of the device. The device ends the service.

6.5.2. End Service of the Mobile Shared Device. The user and the device arrive at a new location, and a new gateway GWN_t is used. The device will end the service after the mutual authentications between the gateway GWN_t and the user u_i as well as the gateway and the device are successful. At the same time, the three parties send the required parameter to the smart contract, and only if all the inputs are matched, the smart contract starts to settle the cost and income of the three parties; else, it returns an error and settlement fails:

E1: $u_i \longrightarrow \text{GWN}_t$: $\text{msg}_{U2} = \{\text{Sig}_{u_i,1}, ID_{u_i}, T_{i2}\}$. The user sends his authentication information to the new gateway. The user u_i sends a request to a new gateway GWN_t nearby to end the use of the device Dev_j. In order to prevent malicious users from pretending himself, the authentication information $\text{msg}_{U2} = \{\text{Sig}_{u_i,1}, ID_{u_i}, T_{i2}\}$ of the user is sent to the gateway

GWN_t, where $\mathrm{Sig}_{u_i,1} = (\sigma_{u_i,1}, L_{i1})$, $\sigma_{u_i,1} = l_i^{-1} \cdot k_{u_i} \cdot h(K_{u_i} \| B\mathrm{add}_{u_i} \| T_{i2}) \bmod p$, $l_{i1} \in Z_p^*$ is selected arbitrarily and kept secretly by the user, and $L_{i1} = l_{i1} \cdot G$ is public.

E2: $\mathrm{GWN}_t \longrightarrow u_i$: msg_{G2}. The gateway sends its own authentication information to the user. After receiving the request, GWN_t checks whether the user is using the device on the blockchain. If so, the gateway authenticates the user with the method in AD3.1. If the authentication is passed, the gateway will send its own authentication information $\mathrm{msg}_{G2} = \{\mathrm{Sig}_{\mathrm{GWN}_t}, ID_{\mathrm{GWN}_t}, T_t\}$ to the user for authentication, where the signature is $\mathrm{Sig}_{\mathrm{GWN}_t} = (\sigma_{\mathrm{GWN}_t}, L_t)$, $\sigma_{\mathrm{GWN}_t} = l_t^{-1} \cdot k_{\mathrm{GWN}_t} \cdot h(K_{\mathrm{GWN}_t} \| ID_{\mathrm{GWN}_t} \| \mathrm{Cert}_{\mathrm{GWN}_t} \| T_t) \bmod p$, $l_t \in Z_p^*$ is selected arbitrarily and kept secretly by the gateway GWN_t, and $L_t = l_t \cdot G$ is public. Then, the gateway GWN_t sends the user's blockchain address $B\mathrm{add}_{u_i}$, the device's identity identifier ID_{Dev_j}, and the termination time to the smart contract.

E3: The user authenticates the gateway as formula (1) in AD3.1. If it passes, the end request message and the gateway's blockchain address $\mathrm{Badd}_{\mathrm{GWN_t}}$ are sent to the smart contract.

E4: The gateway sends its own authentication information msg_{G2} and end instruction to the device $\mathrm{Dev_j}$. If the authentication passes, the device ends the service and passes its own identification identifier $\mathrm{ID}_{\mathrm{Dev_j}}$, termination request, location, status (turned to 0), and the blockchain address $\mathrm{Badd}_{\mathrm{GWN_t}}$ of the new gateway to the smart contract. Otherwise, it does nothing.

E5: After receiving the parameters transmitted by the device, the smart contract calculates the user's expenditure during time $\Delta t = |T_{\mathrm{CurG}} - T_{i1}|$, where T_{CurG} is the time when the new gateway transmitting parameters to the contract. Some service charge is given to the new gateway GWN_t and the balance returns to the user. The specific algorithm of ending service in a smart contract is shown in Figure 4 in the form of a flow chart.

7. Security Analysis

7.1. The Correctness of the Key Negotiation Protocol

7.1.1. The Correctness of Verification Algorithm of the User's Information. If we mark $h_1 = h(K_{u_i} \| ID_{\mathrm{Dev}_j} \| S_i \| R_{u_i} \| T_{i1})$, $h_2 = h(r_i \| k_{u_i} \| B\mathrm{add}_{u_i} \| T_{i1})$, then

$$v_1 = \sigma_{u_i}^{-1} \cdot h_1 \cdot PK_{u_i} = l_i \cdot \frac{h_1 \cdot K_{u_i}}{k_{u_i} h_1 + h_2} = l_i \cdot \frac{h_1 \cdot k_{u_i}}{k_{u_i} h_1 + h_2} \cdot G,$$

$$v_2 = \sigma_{u_i}^{-1} \cdot R_{u_i} = l_i \cdot \frac{h_2}{k_{u_i} h_1 + h_2} \cdot G. \tag{4}$$

And, we can get

$$v_1 + v_2 = l_i \cdot G = L_i. \tag{5}$$

So the verification algorithm is correct.

The correctness proof process of the authentication algorithm of the device is similar to that of the user's, so we omit it here.

7.1.2. The Correctness of the Verification Algorithm of the Gateway. If we mark $h = h(K_{\mathrm{GWN}_k} \| \mathrm{Cert}_{\mathrm{GWN}_k} \| T_{k2})$, then

$$\sigma_{\mathrm{GWN}_k}^{-1} \cdot h \cdot PK_{\mathrm{GWN}_k} = l_k \cdot \frac{h}{k_{\mathrm{GWN}_k} h} \cdot k_{\mathrm{GWN}_k} G = L_k. \tag{6}$$

So the verification algorithm is correct.

7.1.3. The Correctness of the Key Negotiation

$$\begin{aligned} SK' &= h\left(h\left(r_i \| k_{u_i} \middle\| B\mathrm{add}_{u_i} \middle\| T_{i1}\right) \cdot R_{\mathrm{Dev}_j} \middle\| \mathrm{Cert}_{Dev_j} \middle\| B\mathrm{add}_{u_i}\right), \\ &= h\left(h\left(r_i \| k_{u_i} \middle\| B\mathrm{add}_{u_i} \middle\| T_{i1}\right) \cdot h\left(r_j \| k_{\mathrm{Dev}_j} \middle\| ID_{\mathrm{Dev}_j} \middle\| T_{j1}\right) \cdot G \middle\| \mathrm{Cert}_{\mathrm{Dev}_j} \middle\| B\mathrm{add}_{u_i}\right), \\ &= h\left(h\left(r_j \| k_{\mathrm{Dev}_j} \middle\| ID_{\mathrm{Dev}_j} \middle\| T_{j1}\right) \cdot R_{u_i} \middle\| \mathrm{Cert}_{\mathrm{Dev}_j} \middle\| B\mathrm{add}_{u_i}\right), \\ &= SK. \end{aligned} \tag{7}$$

7.2. Security of the Proposed Scheme

7.2.1. Anonymity of Users and Privacy Protection. The blockchain address and the public key are used as the identity information when the user authenticates with the gateway, and the user's blockchain account is used for settlement; these protect the anonymity of the user effectively. Another thing is that the instructions sent to the device by the user are encrypted using symmetric encryption algorithms such as DES, which protects the data confidentiality and prevents attackers from analysing the instructions to obtain private information of the user, such as habits and physical conditions.

7.2.2. User Impersonation Attack. After finishing the prepayment, if the user is replaced by a malicious attacker who wants to replace user's information $\mathrm{msg}_{U1} = \left\{\mathrm{Sig}_{u_i}, K_{u_i}, B\mathrm{add}_{u_i}, ID_{\mathrm{Dev}_j}, S_i, R_{u_i}, T_{i1}\right\}$ with his own information $\mathrm{msg}_{U1} = \left\{\mathrm{Sig}_{A_i}, K_{A_i}, B\mathrm{add}_{A_i}, ID_{\mathrm{Dev}_j}, S_i, R_{A_i}, T_{i1}\right\}$, after receiving the message, the gateway will check whether it

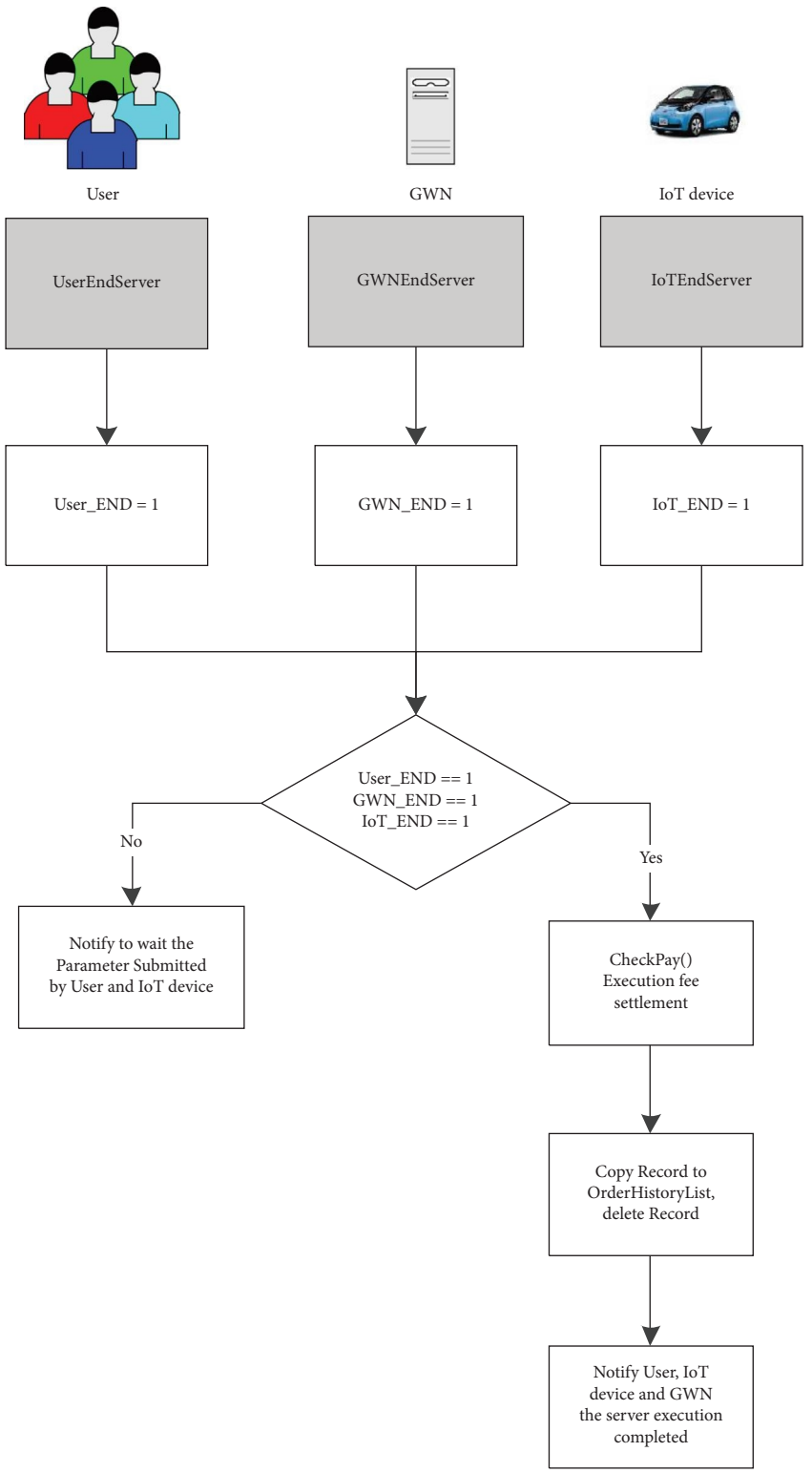

Figure 4: The execution flowchart of ending service in the smart contract. Note: (I) The user calls the interface UserEndServer() to start ending the service process and updates the end of service flag User_END submitted by the user as 1. (II) The GWN calls the interface GWNEndServer() to start ending the service process and updates the end of service flag GWN_END submitted by the GWN as 1. (III) The IoT device calls the interface IoTEndServer() to start ending the service process and updates the end of service flag IoT_END submitted by the IoT device as 1. (IV) It confirms whether all the three parties have submitted the service ending request through the ending service flag when the user, GWN, and IoT device call the process of service ending. If not all the ending requests are submitted, it notifies the user, the GWN, or the IoT device to wait for other parties to submit the ending request within a certain period of time. (V) When all three parties have completed the submission, the expenses shall be settled. And, it records this service as the historical data for follow-up tracking and notifies the user, GWN, and IoT device that the service is finished.

is prepaid successfully by the blockchain address firstly. If it is replaced with the attacker's blockchain address, the verification would fail. If the attacker replaces all other information except the legal user's blockchain address, then the signature verification $\sigma_{A_i} = l_i^{-1} \cdot (k_{A_i} \cdot h(K_{A_i} \| B\text{add}_{u_i} \| ID_{\text{Dev}_j} \| S_i \| R_{A_i} \| T_{i1}) + h(r_i \| k_{A_i} \| B\text{add}_{u_i} \| T_{i1})) \bmod p$ would fail. Because $h(K_{A_i} \| B\text{add}_{u_i} \| ID_{\text{Dev}_j} \| S_i \| R_{A_i} \| T_{i1}) \neq h(K_{u_i} \| B\text{add}_{u_i} \| ID_{\text{Dev}_j} \| S_i \| R_{u_i} \| T_{i1})$, then the equation $v_1 + v_2 = L_i$ will not hold. If the user bribes the gateway and colludes with the gateway, it is impossible for the user and the gateway colluding successfully, and we will analyse it later in the collusion attack.

7.2.3. Gateway Impersonation Attack. The purpose of the malicious gateway A is to earn service fee through submitting its own blockchain address $B\text{add}_{\text{GWN}_A}$ to the smart contract by the user and the device. If the malicious gateway replaces the information of a legal gateway with its own identity information $\text{msg}_{G1} = \{\text{Sig}_{\text{GWN}_A}, ID_{\text{GWN}_A}, T_{k1}\}$, the signature verification will fail. Because it does not have a legal certificate, it is impossible for the malicious gateway to forges a certificate successfully. If the malicious gateway embezzles the certificate of a legal gateway to generate a digital signature and sends $\text{msg}_{G1} = \{\text{Sig}_{\text{GWN}_A}, ID_{\text{GWN}_A}, T_{A1}\}$ to the user, it still cannot make it. When verifying the gateway, the user needs to check the certificate on blockchain firstly; although the certificate provided by the malicious gateway is passed, the verification of the signature $\sigma_{\text{GWN}_A} = l_A^{-1} \cdot k_{\text{GWN}_A} \cdot h(K_{\text{GWN}_A} \| B\text{add}_{\text{GWN}_A} \| ID_{\text{GWN}_A} \| \text{Cert}_{\text{GWN}_K} \| T_{A1}) \bmod p$ fails. That is because the public information of the gateway obtained by the verifier from the blockchain is bound together with the certificate and cannot be tampered with; any changes will led to a failure of signature verification.

7.2.4. Man-in-the-Middle Attack. Man-in-the-middle attacks may occur in some key negotiation schemes. In our scheme, two random numbers r_i and r_j are set for the user and the device when they calculate a session key. The two random numbers are kept secretly by the user and the device, respectively. In addition, when the authentication messages are transmitted by the two parties, there are signatures on the random numbers; as a result the intermediary cannot forge signatures to negotiate the key with the user and the device.

7.2.5. Collusion Attack. The user colludes with the gateway. It is impossible for the user to collude with the gateway successfully and access the device without prepayment, for the smart contract will check the prepayment firstly when arriving to the last step in the key negotiation phase. If the gateway and a device collude, the legal device is replaced with an illegal one. While the user has checked the device's ID on the blockchain before prepayment, the prepayment address is determined by the user, so the device cannot be changed.

7.2.6. Denial of Service Attack. Since the blockchain and the gateway are composed of multiple nodes and they are distributed, a denial of service attack on one node can be achieved, but in a distributed system, it is difficult to achieve a denial of service attack.

7.2.7. Refuse to Replay Attack. A time stamp is added to the authentication messages, in order to verify the freshness of the message and prevent replay attacks.

7.2.8. Reliability of Service. We store some key identity information on the blockchain; as a result, the attacker ccannot tamper with them and other secret information is protected by a signature. The smart contract can be executed automatically without human intervention and is more fair and just. If the user prepays, then he can get the service he wants.

8. Performance Evaluation and Simulation Results

In this section, we compare the realized functionalities in the proposed scheme with relevant schemes firstly. Then, the communication costs of the proposed scheme at different phases are analysed and compared with two schemes. The time costs of proposed scheme are tested in the simulation experiment from different concerns and compared with the relevant schemes. Finally, the gas costs of the smart contract are tested on the Ethereum test network.

Experimental setup: In order to verify the proposed scheme, we carried out 50 repetitions of the experiment. The experimental setup contained the personnel computer with 3.20 GHZ Inter(R), Core(TM) i7-8700 CPU@3.20 GHz, and RAM16.0 GB as a gateway and a CA. The operating system is Windows10. And, a mobile phone having a 2.45 G processor and 2 GB memory is regarded as a user and the shared IoT device is set as a Raspberry pi3 B+ with 1.4 GHZ CPU and 1 GB DDR2. The PBC library is called for elliptic curve cryptography and pair-based cryptography.

8.1. Comparison of Functions. Table 2 shows some functions about access control schemes in the IoT environment. From Table 2, we can see that our scheme achieves a higher security goal under a weaker security assumption while having a decentralized feature. In these functions, the gateway being not secure and support mobility are two basic requirements in shared IoT device circumstances, so we are going to compare our scheme with references [17, 18] in the next sections.

8.2. The Analysis of Communication Efficiency. The communication cost is an important indicator to evaluate the efficiency of a scheme [14–18]. The lower the communication cost, the higher the efficiency of the scheme. In our assessment, we use the following security parameters: $|Z_q^*|$ is the length of an element in Z_q^*, and it is 1024 bit. The private key of the elliptic curve is a positive integer whose length is

Table 2: The summary of functions about some access control schemes in the IoT environment.

Schemes	FSA1	FSA2	FSA3	FSA4	FSA5	FSA6	FSA7
[11]	×	×	✓	×	×	×	×
[17]	✓	×	×	×	✓	×	✓
[18]	✓	×	×	✓	✓	✓	✓
[23]	✓	×	✓	✓	✓	✓	×
[25]	✓	×	✓	×	×	×	×
[26]	✓	×	×	✓	×	✓	×
[29]	✓	×	✓	✓	✓	✓	×
[30]	✓	×	✓	×	×	×	×
[31]	×	✓	✓	×	✓	×	×
Ours	✓	✓	✓	✓	✓	✓	✓

FSAi ($i = 1, \ldots, 7$) is the ith attribute. FSA1: whether mutual authentications occur with each other; FSA2: no need of a secure channel; FSA3: is it decentralized; FSA4: whether the key is negotiated; FSA5: whether to support mobility; FSA6: does the privacy be protected; FSA7: the gateway is not secure.

160 bits, and the X-coordinate and Y-coordinate of the elliptic curve point take 160 bits, respectively. And, 160-bit ECC security remains same as that for a 1024-bit RSA public key cryptosystem [40]. And, the bilinear pairing operation and modular exponential calculation require 1024 bits to achieve the same security level as the elliptic curve. The output of the hash is 160 bits, the length of an ID is 32 bits, the time stamp is 32 bits, and the blockchain address is 272 bits.

The scheme includes five phases: system initialization, entity registration, security channel establishment, service process, and end of service. Since the phases of system initialized and entity registration are offline, we will not analyse the communication cost at these phases. In the service process, the user's instructions are the same under the same circumstances, so we only compare the communication costs of different schemes in the security channel establishment phase and the end of the service phase.

The message streams sent by different entities in different schemes are as follows.

8.2.1. Secure Channel Establishment Phase. In Liu's scheme, the message sent by the user is $\eta \in Z_q^*$, E_{sk}(cmd, Parm, ID_j, pid), σ, R and the total communication cost is $1024 + 1024 + 1024 + 1024 = 4096$ bits. The message sent by the gateway is $\{k_2, \beta\}$, $\{\sigma, \theta, SL\}$, and $\sigma = \langle R_i, D_i, Z, TS_2 \rangle$, so the total size is $1024 + 160 + 1024 + 1024 + 1024 + 32 + 17 + 32 * 3 = 4401$ bits. The device passes $\{y, TS_j, \text{profile}_j\}$ and $\{S\}$, so the total size is $1024 + 32 + 32 + 1024 = 2112$ bits.

In LIAP, the message sent by the user is $M_r = (PK, \text{cert}, \sigma, T)_{PK_{Ri}}$ and $(\text{PID}_i, M_s, PK_{Ri}, \sigma_i)$, so the total communication is $92 * 8 + 416 + 32 + 160 + 320 * 2 = 1984$ bits. And, the data packet sent by the gateway in LIAP is $M_h = (PK_{Ri}, \text{cert}_{Ri}, Te, \text{RPK}_i^1, \text{RPK}_i^2, \text{RPK}_{i-1}^1, \text{RPK}_{i-1}^2, \text{RPK}_{i+1}^1, \text{RPK}_{i+2}^2)$, σ_{Ri}, and $(\text{Cert}_{Ri}, m_i^1, m_i^2, \text{sign}(SK_{Ri}, H(\text{Cert}_{Ri}, m_i^1, m_i^2)))_{PK_{Ri}}$, so the communication cost is $160 + 160 + 168 + 32 + 160 * 6 + 21 * 8 + 92 * 8 = 2384$ bits. The device pass $M_r = (PK, \text{cert}, \sigma, T)_{PK_{Ri}}$ and $(\text{PID}_i, M_s, PK_{Ri}, \sigma_i)$, and the total size is $92 * 8 + 416 + 32 + 160 + 320 * 2 = 1984$ bits.

In the proposed scheme, the message flow sent by the user is $\text{msg}_{U1} = \{\text{Sig}_{u_i}, K_{u_i}, B\text{add}_{u_i}, ID_{\text{Dev}_j}, S_i, R_{u_i}, T_{i1}\}$, a contract address of payment success to service S_i and $\{\text{SKV}, B\text{add}_{\text{GWN}_k}\}$. So the total communication cost is $320 + 160 + 272 + 32 + 32 + 160 + 32 + 272 + 160 + 272 = 1712$ bits. The messages sent by the gateway are $g'_{U1} = \{\sigma_{u_i}, K_{u_i}, B\text{add}_{u_i}, ID_{\text{Dev}_j}, S_i, R_{u_i}, T_{i1}\}$, twice $\text{msg}_{G1} = \{\text{Sig}_{\text{GWN}_k}, ID_{\text{GWN}_k}, T_{k1}\}$ and $\text{msg}'_{D1} = \{\sigma_{\text{Dev}_j}, ID_{\text{Dev}_j}, R_{\text{Dev}_j}, T_{j1}\}$, so the communication cost of the gateway is $320 + 160 + 272 + 32 + 32 + 160 + 32 + 2(160 + 160 + 32) + 160 + 32 + 160 + 32 = 2096$ bits. The message sent by the device includes $\text{msg}_{D1} = \{\text{Sig}_{\text{Dev}_j}, ID_{\text{Dev}_j}, R_{\text{Dev}_j}, T_{j1}\}$, SKV, and $B\text{add}_{\text{GWN}_k}$, so the communication cost is $320 + 32 + 160 + 32 + 160 + 272 = 976$ bits.

8.2.2. End of Service Phase. When the stationary device is terminated, it does not need a new gateway, and the user only needs to send the termination instruction 0 directly, so we will not discuss it. Here, we mainly discuss the communication cost of the dynamic device. And, service termination is not involved in the scheme LIAP [17], so we only compare it with Liu et al. 's scheme [18].

In Liu's scheme, the user need passes Rtn, $(\sigma, \theta, \eta), ID_i, \eta'$ and $R_{\text{new}}', \sigma', (\sigma_1', \ldots, \sigma_{\text{num}}'), C$, to the gateway, that is $(1024 + 1024 + 160 + 32) * 2 + 17 + 1024 + 32 + 1024 + 1024 + 256 = 5617$ bits, and the gateway need pass $(\theta_1, \ldots, \theta_{\text{num}}), k_2, \beta, \sigma_{\text{new}}, \text{num},$; the data size is $1024 + 1024 + 160 + 32 + 1024 + 1024 + 160 = 4448$ bits; here, we only record one OTS. The IoT device transmits $\{y, TS_j, \text{profile}_j\}$ and $\{S\}$, which is $1024 + 32 + 32 + 1024 = 2112$ bits.

In the proposed scheme, the data message sent by the user is $\text{msg}_{U2} = \{\text{Sig}_{u_i,1}, ID_{u_i}, T_{i2}\}$ and a termination instruction 0; the communication cost is $320 + 32 + 32 + 1 = 385$ bits. The message sent by the gateway is twice $\text{msg}_{G2} = \{\text{Sig}_{\text{GWN}_t}, ID_{\text{GWN}_t}, T_t\}$, $B\text{add}_{u_i}$, ID_{Dev_j}, and T_t, so the communication cost of the gateway is $2(320 + 32 + 32) + 272 + 32 + 32 = 1104$ bits, and the device sent messages ID_{Dev_j}, $B\text{add}_{\text{GWN}_t}$, and termination request 0, so the total communication cost is $32 + 272 + 1 + 32 = 337$ bits.

Figures 5 and 6 show the communication costs of three entities including the user, gateway, and device in contrast

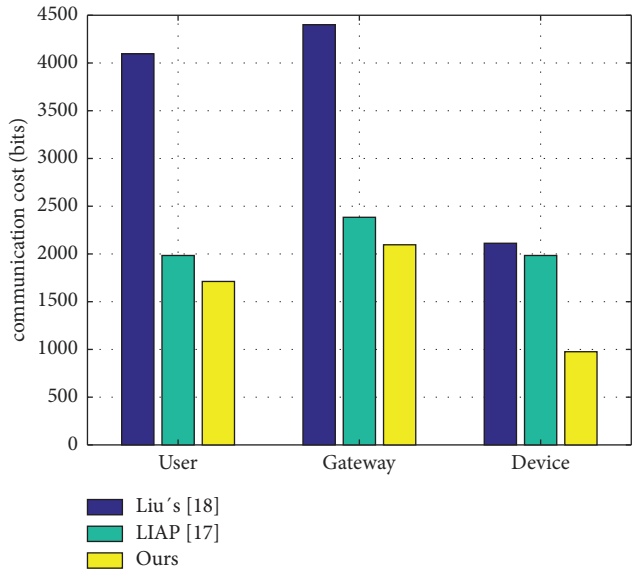

Figure 5: Comparison of communication cost of the three entities in different schemes in the secure channel establishment phase.

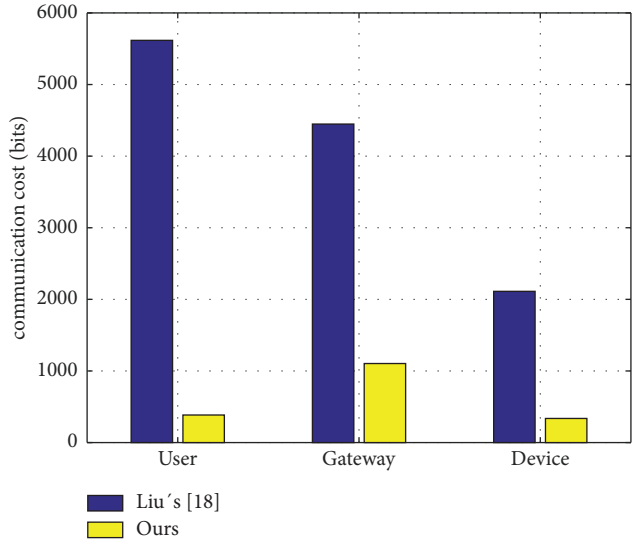

Figure 6: Comparison of communication cost of the three entities in our scheme and Liu's scheme in the end of service phase.

schemes and ours. From Figures 5 and 6, we can see that the communication costs of the user, gateway, and device are relatively low in the security channel establishment phase and end of service phase in our scheme. In particular, the communication cost of our scheme on the user side and device side is less than 50% compared with Liu et al. 's [18] scheme. Another thing is because of the use of smart contracts, the communication cost of our scheme in the end of service phase is far less than that of Liu et al. 's [18] scheme. Therefore, the communication efficiency of our scheme is higher. And, as far as our scheme is concerned, because some tasks of users and devices are assigned to the gateway, we can see that the user and device with weaker communication capabilities undertake lower communication burden than that of the gateway.

8.3. The Analysis of Time Cost. The time cost is another important indicator to evaluate the efficiency of the scheme [14–18]. Obviously, the less the time cost, the higher the efficiency of a scheme. Similarly, we evaluate the time cost in these three phases: entity registration, security channel establishment, and end of service.

The detail computation operation performed by various components of different scheme is listed in Tables 3–5 in different phases. We use the $T_{bp}, T_{mtp}, T_e, T_{sm}, T_h$, T_{OTS}, and T_{vOTS} to represent the operation of a bilinear pairing, a map-to-point hash function, an exponential operation on G_1, a scalar multiplication on elliptic curve, a general hash function operation, one-time signature, and the authentication of one-time signature, respectively. For the operations take very little time, such as the addition of numbers, we ignore them. In particular, we can see from Table 4 that because users and devices outsource verification tasks to the gateway, their computation burden is reduced by about 50%.

In order to evaluate the time costs on computation, we tested 50 times in our simulation environment and the results are shown in Figure 7 (in the registration phase), Figure 8 (security channel establishment phase), and Figure 9 (end of service phase of dynamic device). They mainly show the time costs of the entities including the user, gateway, and device in different phases in references [17, 18] and our scheme.

From Figure 7, we can see that the time costed by the gateway in the registration phase is little in our scheme, while the registration phase is as a preparation work; it will not be executed during real-time access control. When a user accessing and using a shared device in real time, it is only related to the secure channel establishment phase and end of service phase, and these time costs show in Figures 8 and 9. We can see the three entities all cost less time in our scheme than that in references [17, 18]. In the secure channel establishment phase, the user costs about 2.5 ms, and in Liu's scheme, a user cost is about 35.7 ms and 21.3 ms in LIAP; the user of our scheme has much less computation cost than that of others and the same as the entities of the gateway and device. In LIAP, the authors did not consider the situation of end of service, so we run Liu's scheme and ours; the result shows the time costs on computation of different entities in our scheme are much less than those of Liu's.

Therefore, our scheme is more suitable for IoT devices with low computing and communication power. Especially in the environment of a user to use a shared IoT device, the efficient communication and computing efficiency can increase the experience effect of users.

The total time cost is the focus of users. When multiple users apply for access IoT devices at the same time, we conducted tests on the time cost off-chain. Experiment

Table 3: The computation cost in the registration phase.

Schemes	User/device	Gateway	Device	CA
Liu et al. 's [18]	0	0	0	$3T_e$
LIAP [17]	0	0	$2T_{sm}$	$4T_{sm}$
Proposed scheme	0	$2T_{sm}$	$2T_{sm}$	$2T_{sm}$

Table 4: The computation cost in the authentication phase.

Schemes	User/device	Gateway	Device
Liu et al. 's [18]	$5T_e + 2T_h$	$6T_e + 2T_{bp} + T_h$	$5T_e + 4T_h$
LIAP [17]	$2T_{bp} + 2T_{sm} + T_{mtp} + 3T_e$	$3T_{sm} + 3T_e$	$2T_{bp} + 2T_{sm} + T_{mtp} + 3T_e$
Proposed scheme	$2T_{sm} + 3T_h$	$5T_{sm} + 5T_h$	$2T_{sm} + 5T_h$

Table 5: The computation cost of dynamic device in the end of service phase.

Schemes	User	Gateway	Device
Liu et al. 's [18]	$5T_e + 2T_h + mT_{\text{OST}}$	$2T_{bp} + 9T_e + 6T_h + mT_{v\text{OTS}}$	$5T_{bp} + 2T_h$
Proposed scheme	$2T_{sm} + 2T_h$	$2T_{sm} + 2T_h$	$T_{sm} + T_h$

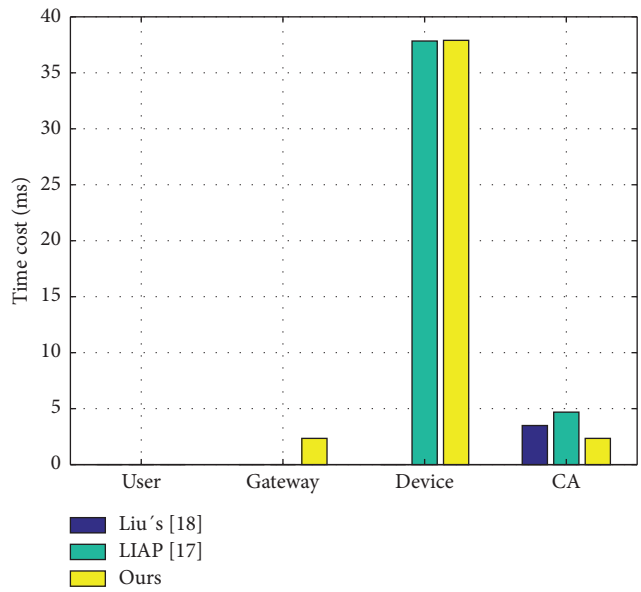

Figure 7: Comparison of time cost on computation of four entities in the registration phase in different schemes.

content: We simulate the process of users accessing the IoT devices, and the reply of blockchain is true by default. We simulate the concurrency experiment off-chain. Users and IoT devices are all executed alone; concurrency is not involved. The concurrent operation only involves the gateway and blockchain. Therefore, we simulate the concurrency in the gateway and execute N groups of tasks. The user and the IoT device execute a real process of secure channel establishment and end of service, and the real task is inserted at the end of the task queue. When carrying out the task, the gateway simulates the blockchain to return true. An experimental framework is shown in Figure 10.

The experimental results are shown in Figure 11. As shown in Figure 11, with the increase in users' number, the time costs of the start access and end access do not increase proportionally. When the number of concurrent request services is 4 and 8, the time cost increases in a step growth. In other cases, the image is gentle and the time is hardly increased. The tasks are processed by multicore and multithread in the gateway. If the configuration of the current gateway is in use, the time cost increases separately as 8 ms and 2 ms in the establishment phase and end of service phase to every additional 4 users. Therefore, when the number of users is no more than 100 in the same gateway, the user's application response delay will be less than 800 ms in the establishment phase and less than 200 ms in the end of service phase. Especially, when the service is ended, the time cost by the users is much less than

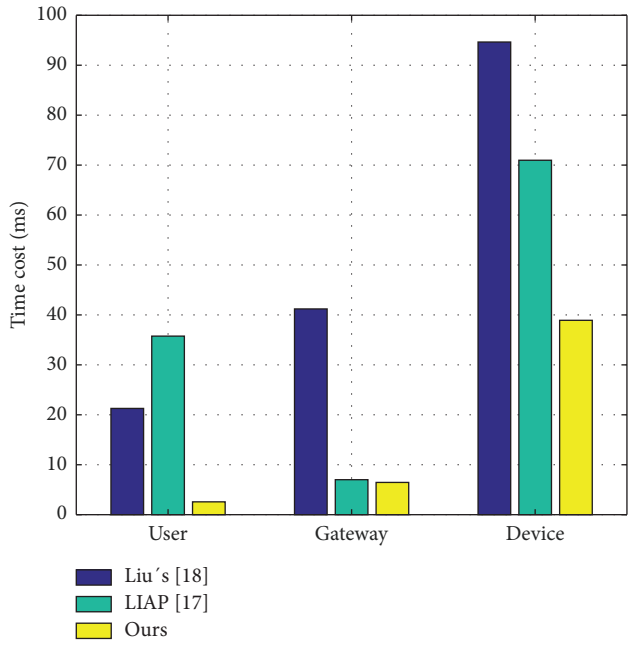

Figure 8: Comparison of time cost on computation of three entities in the secure channel establishment phase in different schemes.

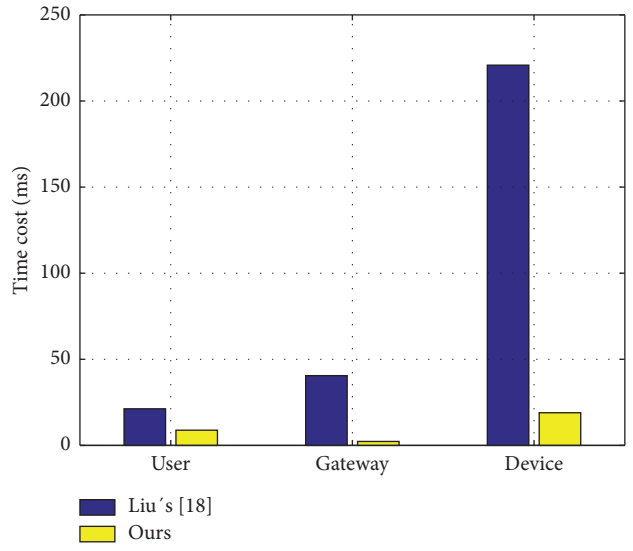

Figure 9: Comparison of computation cost of dynamic device in the end of service phase.

that of the access establishment phase and the end of the access can be achieved quickly. It will not affect experience of users.

8.4. The Analysis of Smart Contracts. Because there is no smart contract in the literature [17, 18], we only test the efficiency of our smart contract. In this section, we test the deployed smart contract and record the gas cost at different phases. The gas cost can evaluate the performance of a smart contract; the less the gas cost, the higher the efficiency of a smart contract. Figures 12 and 13 show the execution efficiency of our smart contract, which list the gas costs of different operations. The smart contracts are tested on the Ethereum testnet (https://remix.ethereum.org/). Because the smart contract we designed only has judgment statements, it submits and queries operations, making it cost relatively less gas [41, 42]. According to literature [42], the gas required of general smart contract is millions or more, while our gas cost is about one tenth of that of general smart contracts.

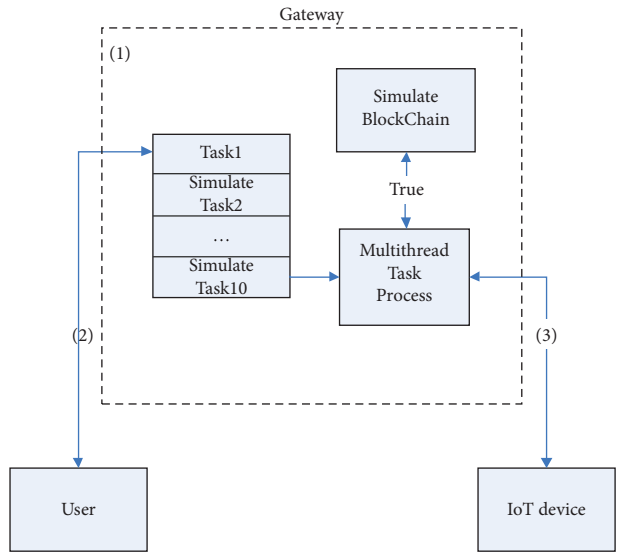

Figure 10: Experimental architecture of concurrence user access. Note: (1) The gateway builds simulate tasks and execute tasks. (2) The interactions between the gateway and user. (3) The interactions between the gateway and IoT device.

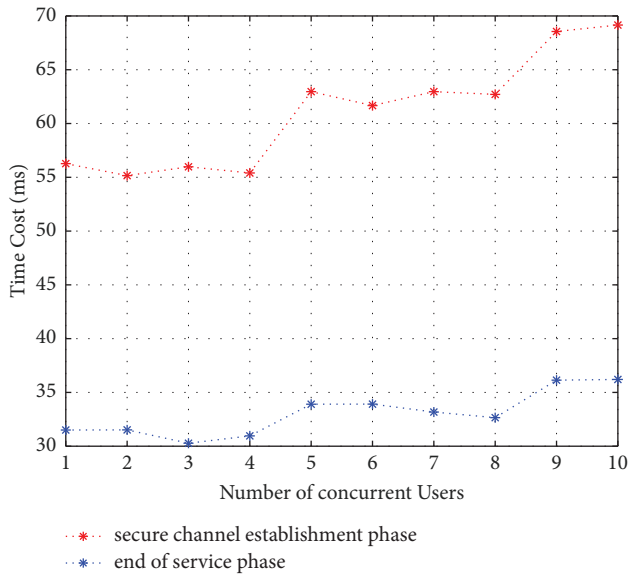

Figure 11: Time cost of users accessing devices off-chain in our scheme.

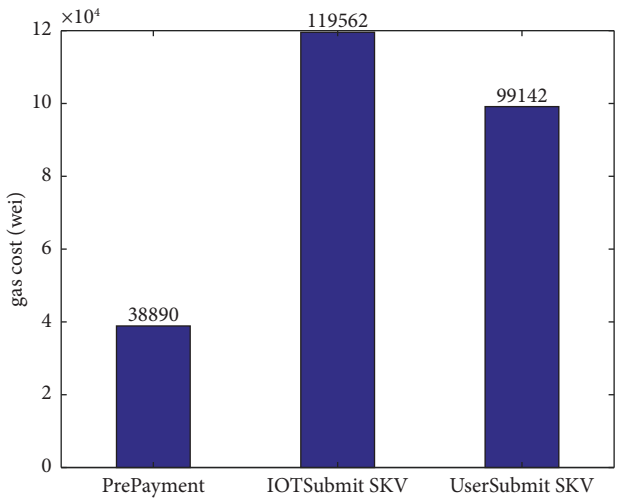

Figure 12: The gas cost during the secure channel establishment phase.

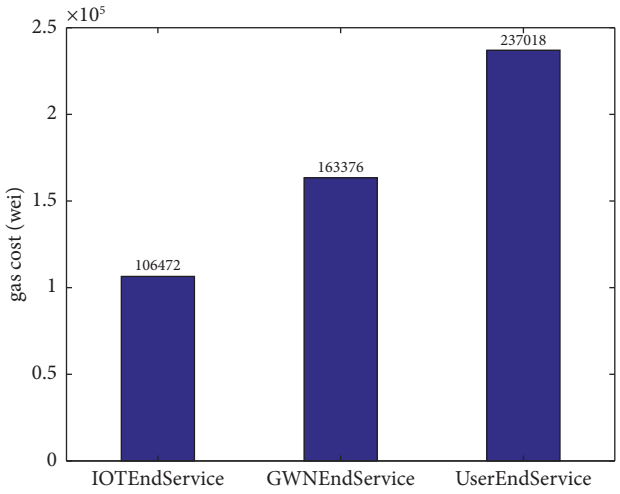

Figure 13: The gas cost in the end of service phase.

9. Conclusion

Considering the distributed characteristic of each entity in a shared IoT scenario and the low communication and computing power of the shared IoT devices, a decentralized real-time access control scheme is utilized to realize the use and the end of shared IoT devices by any users. The use of smart contract prevents malicious behaviours such as centralized corruption and tampering with transaction flows effectively. The scheme achieves access rights of legitimate users effectively and prevents malicious behaviours of illegal users, devices, and gateways. The payment function of the blockchain solves the authority problem of using the device from users perfectly and avoids the use of overly complex public key cryptography technology. An elliptic curve-based authentication protocol completes access control together with the smart contract and protects users' privacy, which is suitable for IoT devices and users. Experimental analysis shows calculation burden undertaken by each party is reasonable and efficient, the amount of communication is relatively small, and the shared service contract deployment is reasonable and efficient. The efficiency of the traditional single-chain blockchain needs to be improved, and in the future, we plan to use the DAG blockchain to improve transaction efficiency or use the thought of shard to improve the efficiency of the system.

Data Availability

The software codes used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61572019), the Key Research and Development Program of Shaanxi (Grant no. 2020GY-006), and Science and Technology Project of Shaanxi Province (Project no. 2022GY-040).

References

[1] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: current status and key issues," *International Journal on Network Security*, vol. 3, no. 2, pp. 101–115, 2006.

[2] S. Luo, J. Hu, and Z. Chen, "An identity-based one-time password scheme with anonymous authentication," in *Proceedings of the International Conference on Networks Security Wireless Communications and Trusted Computing*, pp. 864–867, Wuhan, China, April 2009.

[3] J. Y. Liu, A. M. Zhou, and M. X. Gao, "A new mutual authentication scheme based on nonce and smart cards," *Computer Communications*, vol. 31, no. 10, pp. 2205–2209, 2008.

[4] S. Matsumoto and R. M. Reischuk, "IKP: turning a PKI around with decentralized automated incentives," in *Proceedings of the Process of 2017 IEEE Symposium on Security and Privacy*, pp. 410–426, IEEE, San Jose, CA, USA, May 2017.

[5] G. Rathee, A. Sharma, R. Kumar, and R. Iqbal, "A secure communicating things network framework for industrial IoT using blockchain technology," *Ad Hoc Networks*, vol. 94, Article ID 101933, 2019.

[6] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent advances in smart contracts: a technical overview and state of the art," *IEEE Access*, vol. 8, pp. 117782–117801, 2020.

[7] M. Poongodi, A. Sharma, V. Vijayakumar et al., "Prediction of the price of ethereum blockchain cryptocurrency in an industrial finance system," *Computers & Electrical Engineering*, vol. 81, Article ID 106527, 2020.

[8] H. H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1005–1023, 2014.

[9] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, and I. Verbauwhede, "SPONGENT: the design space of lightweight cryptographic hashing," *IEEE Transactions on Computers*, vol. 62, no. 10, pp. 2041–2053, 2013.

[10] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.

[11] A. Y. F. Alsahlani and A. Popa, "LMAAS-IoT: lightweight multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment," *Journal of Network and Computer Applications*, vol. 192, Article ID 103177, 2021.

[12] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the Internet of Things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015.

[13] J. Granjal, E. Monteiro, and J. Sa Silva, "Security for the Internet of Things: a survey of existing protocols and open

research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[14] S. Banerjee, V. Odelu, A. K. Das et al., "A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8739–8752, 2019.

[15] H. Abdi Nasib Far, M. Bayat, A. Kumar Das, M. Fotouhi, S. M. Pournaghi, and M. A. Doostari, "LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT," *Wireless Networks*, vol. 27, no. 2, pp. 1389–1412, 2021.

[16] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9762–9773, 2019.

[17] S. Wang and N. Yao, "LIAP: a local identity-based anonymous message authentication protocol in VANETs," *Computer Communications*, vol. 112, pp. 154–164, 2017.

[18] Y. Liu, K. Xue, P. He, D. S. L. Wei, and M. Guizani, "An efficient, accountable, and privacy-preserving access control scheme for internet of things in A shared economy environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 99–112, 2020.

[19] H. Du, Q. Wen, S. Zhang, and M. Gao, "A new provably secure certificateless signature scheme for Internet of Things," *Ad Hoc Networks*, vol. 100, Article ID 102074, 2020.

[20] S. Tangade, S. S. Manvi, and P. Lorenz, "Decentralized and scalable privacy-preserving authentication scheme in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8647–8655, 2018.

[21] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[22] G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, "A blockchain framework for securing connected and autonomous vehicles," *Sensors*, vol. 19, no. 14, 2019.

[23] Z. Xu, W. Liang, K.-C. Li, J. Xu, and H. Jin, "A blockchain-based Roadside Unit-assisted authentication and key agreement protocol for Internet of Vehicles," *Journal of Parallel and Distributed Computing*, vol. 149, no. 6, pp. 29–39, 2021.

[24] A. Sharma, R. Tomar, N. Chilamkurti, and B. G. Kim, "Blockchain based smart contracts for internet of medical things in e-healthcare," *Electronics*, vol. 9, no. 10, pp. 1609–1622, 2020.

[25] Q. Fan, J. Chen, L. J. Deborah, and M. Luo, "A secure and efficient authentication and data sharing scheme for Internet of Things based on blockchain," *Journal of Systems Architecture*, vol. 117, Article ID 102112, 2021.

[26] Z. Cui, F. Xue, S. Zhang et al., "A hybrid BlockChain-based identity authentication scheme for multi-WSN," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 241–251, 2020.

[27] C. Zhang, L. Zhu, and C. Xu, "BPAF: blockchain-enabled reliable and privacy-preserving authentication for fog-based IoT devices," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 88–96, 2022.

[28] G. Rathee, A. Sharma, H. Saini, R. Kumar, and R. Iqbal, "A hybrid framework for multimedia data processing in IoT-healthcare using blockchain technology," *Multimedia Tools and Applications*, vol. 79, no. 15-16, pp. 9711–9733, 2020.

[29] M. S. Eddine, M. A. Ferrag, O. Friha, and L. Maglaras, "EASBF: an efficient authentication scheme over blockchain for fog computing-enabled internet of vehicles," *Journal of Information Security and Applications*, vol. 59, no. 3, Article ID 102802, 2021.

[30] L. Vishwakarma and D. Das, "SCAB-IoTA: secure communication and authentication for IoT applications using blockchain," *Journal of Parallel and Distributed Computing*, vol. 154, no. 4, pp. 94–105, 2021.

[31] R. Lavanya, K. Sundarakantham, S. M. Shalinie, R. Divya, and S. Selvamani, "User authentication of iot devices for decentralized architecture using blockchain," in *Proceedings of the International Conference On Advanced Communication And Networking*, pp. 15–26, Singapore, Asia, July 2019.

[32] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: a decentralized blockchain-based authentication system for IoT," *Computers & Security*, vol. 78, pp. 126–142, 2018.

[33] D. Li, Y. Jia, X. Gao, and N. Al-Nabhan, "Research on mulitidomain authentication of IoT based on cross-chain technolgy," *Security and Communication Networks*, vol. 2020, Article ID 6679022, 12 pages, 2020.

[34] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," in *Proceedings of the 2018 IEEE/ACA 15th International Conference on Computer Systems and Applications (AICCAA)*, pp. 1–8, Aqaba, Jordan, October 2018.

[35] G. Mwitende, Y. Ye, I. Ali, and F. Li, "Certificateless authenticated key agreement for blockchain-based WBANs," *Journal of Systems Architecture*, vol. 110, Article ID 101777, 2020.

[36] C. Diem, "On the discrete logarithm problem in elliptic curves," *Compositio Mathematica*, vol. 147, no. 1, pp. 75–104, 2010.

[37] S. Underwood, "Blockchain beyond Bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016.

[38] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: securing a blockchain applied to smart contracts," in *Proceedings of the 2016 ICCE*, pp. 467-468, Las Vegas, NV, USA, March 2016.

[39] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 254–269, Vienna, Austria, October 2016.

[40] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for Internet of Drones," *Computer Communications*, vol. 154, pp. 455–464, 2020.

[41] A. Saini, Q. Zhu, N. Singh, Y. Xiang, L. Gao, and Y. Zhang, "A smart-contract-based access control framework for cloud smart healthcare system," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5914–5925, 2021.

[42] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.