

Received April 3, 2020, accepted April 27, 2020, date of publication May 4, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992385

# Smart Contract-Based Spectrum Sharing Transactions for Multi-Operators Wireless Communication Networks

SHUANG ZHENG<sup>ID</sup>, (Student Member, IEEE), TAO HAN<sup>ID</sup>, (Member, IEEE),

YUNA JIANG<sup>ID</sup>, (Student Member, IEEE), AND

XIAOHU GE<sup>ID</sup>, (Senior Member, IEEE)

School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Tao Han (hantao@hust.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2020kfyXGYJ115.

**ABSTRACT** Multiple-operators (multi-OPs) spectrum sharing mechanism can effectively improve the spectrum utilization in fifth-generation (5G) wireless communication networks. The secondary users are introduced to opportunistically access the licensed spectrum of idle operators (OPs). However, the identity privacy and data security issues raise great concerns about the secure spectrum sharing among multi-OPs. To address these challenges, a permissioned blockchain trust framework is proposed for the spectrum sharing in multi-OPs wireless communication networks in this paper. The Multi-OPs Spectrum Sharing (MOSS) smart contract is designed on the constructed permissioned blockchain to implement the spectrum trading among multi-OPs. Without the need of trustless spectrum broker, the MOSS smart contract enforces multi-OPs to share the spectrum truthfully and designs a punishment mechanism to punish malicious OPs. The MOSS smart contract is tested on the Remix integrated development environment (IDE) and the gas costs of MOSS smart contract is estimated. The performance analysis of the proposed permissioned blockchain trust framework demonstrates that the privacy, openness and fairness of the proposed solution are better than traditional spectrum allocation solutions.

**INDEX TERMS** Spectrum sharing, permissioned blockchain, smart contract, privacy, security.

## I. INTRODUCTION

Wireless spectrum, as a scarce natural resource, is an essential foundation to support wireless communications. The exclusive spectrum is statically allocated to an operator (OP) in traditional spectrum management solutions. Based on the statistics in the Federal Communications Commission's investigation report [1], the utilization of the licensed spectrum ranges from 15% to 85% with traditional static spectrum allocation solutions. The communication traffic has exploded with the rise of high definition (HD) video transmission, virtual reality (VR), augmented reality (AR) and other high-bandwidth services in fifth-generation (5G) wireless communication networks [2]. The traditional static spectrum allocation and management solutions would cause the spectrum underutilization and inefficiency, which have great limitations on meeting traffic needs of 5G wireless communication networks. Therefore, it is urgent to find new

solutions to improve the spectrum utilization for the spectrum management and allocation in wireless communication networks.

Dynamic spectrum sharing in cognitive radio (CR) networks is a feasible solution to relieve the shortage and underutilization of spectrum resources [3]–[5]. CR is defined as an intelligent wireless communication system which can sense and learn from the surrounding environment by the understanding-by-building method [4]. Based on the cognitive and reconfigurable abilities of CR, primary users (PUs) can share its licensed spectrum with secondary users (SUs) to improve the spectrum utilization [6]. The microeconomic theories, such as the auction theory [7], the game theory [8], the contract theory [9], [10] and the price theory [11], were adopted to realize the dynamic spectrum sharing between PUs and SUs. Two main architectures have been proposed in previous studies of dynamic spectrum sharing among multi-operators (multi-OPs): the centralized [12] and the distributed [13] spectrum sharing architectures. Although both architectures can relieve the spectrum shortage and improve

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan Shahid<sup>ID</sup>.

the spectrum utilization, security issues of interaction among mutual-trustless entities are severe and have not been considered in previous studies. In the centralized spectrum sharing architecture, a spectrum broker is introduced to be responsible for the spectrum allocation between participating PUs and SUs. In order to share the spectrum efficiently, both PUs and SUs need to interact the private information with the trustless spectrum broker [14]. In this case, the interaction suffers from a single-point failure and privacy disclosure. In the distributed spectrum sharing architecture, participating PUs and SUs need to interact with each other [15], which may face the challenge of huge communication cost and privacy leakage. Therefore, it is necessary to propose an efficient solution to guarantee the secure interaction in the dynamic spectrum sharing among multi-OPs.

Recently, the promising blockchain [16] and smart contract [17] technologies have been widely used in many fields due to its decentralization and security, such as the medical treatment [18], the big data [19], the electrical vehicles [20], and the Internet of things (IoT) [21], etc. The smart contract consists of many predefined functions which can be triggered by transactions to realize specific functionalities. Transactions are recorded on the distributed ledger maintained by blockchain consensus nodes. The blockchain has also been applied to the spectrum sharing with advantages of security, fair and transparency. The ownership and usage of spectrum are recorded on the distributed ledger of blockchain [22], [23]. The security and privacy-preserving of spectrum sharing among multi-OPs can be guaranteed without the existence of a trustless spectrum broker [24], [25]. However, due to the high block verification delay and low throughput of the public blockchain, the efficient spectrum sharing cannot be achieved in existing blockchain studies.

To overcome these issues, in this paper, a permissioned blockchain is constructed for the spectrum sharing in multi-OPs wireless communication networks. The practical byzantine fault tolerant (PBFT) consensus algorithm instead of proof of work (POW) is adopted in the permissioned blockchain, which can greatly realize the the high throughput and short delay. The multi-OPs acted as authorized nodes are required to be authenticated the certificate by the administrator before joining in the constructed permissioned blockchain. In this paper, the Multi-OPs Spectrum Sharing (MOSS) smart contract is designed on the blockchain for spectrum sharing in wireless communication networks. Without a trustless spectrum broker, different OPs can autonomously trade the spectrum by calling functions defined in the MOSS smart contract.

The main contributions in this paper are described as follows.

- 1) Based on the constructed permissioned blockchain, a decentralized and secure framework is proposed for the spectrum sharing in multi-OPs wireless communication networks.
- 2) The double auction and free-trading market are introduced to design the MOSS smart contract, which

enables multi-OPs to autonomously share the spectrum in wireless communication networks.

- 3) The MOSS smart contract is compiled on the Remix integrated development environment (IDE). The gas cost of each function defined in the MOSS smart contract is estimated. The performance analysis of the proposed spectrum sharing solution are given.

The rest of this paper is organized as follows. In Section II, related works are firstly discussed. In Section III, a permissioned blockchain trust framework is proposed for the spectrum sharing in multi-OPs wireless communication networks. In Section IV, the MOSS smart contract is designed. In Section V, experimental results are presented. The performance analysis are also discussed. Finally, Section VI concludes the paper.

## II. RELATED WORK

In this section, related works about the spectrum sharing with the CR technology are discussed, and the existing studies of the spectrum sharing with the blockchain technology are also introduced.

Many studies combine the CR technology with the microeconomic model to solve the challenges of spectrum shortage and underutilization. The auction-based approach was designed in [7] to solve the spectrum sharing between macro base stations (MBS) and femto access points (FAP), who acted as the bidder for the additional spectrum of MBSs. The many-to-one stable matching game theory and the stochastic geometrical approaches were used in [8] to tackle the dynamic spectrum sharing among network OPs. The contract and price theory were also applied to the spectrum sharing in CR networks. Considering the scenario consisted of a single PU and multi-SUs, an optimal contract was derived to satisfy the incentive compatibility and individual rationality [9]. A hybrid distributed-centralized spectrum sharing framework was proposed in [10] based on the contract theory, where a random leader mechanism was introduced to randomly elect the spectrum manager among SUs or PUs. By using the Equilibrium price theory, a distributed algorithm based on the Bertrand game and repetitive game was proposed to maximize the benefits of multi-PUs and satisfy the demands of SUs [11]. However, these studies did not consider the security issue in the spectrum sharing among multi-OPs, which may cause the privacy disclosure of OPs.

The emergence of blockchain technology provides a new way to ensure the security and decentralization. There exist some studies applying the blockchain and smart contract for various applications. A healthcare information exchange blockchain platform was built in [18], which combined the off-chain storage and on-chain verification to ensure the privacy and authentication of healthcare data. An energy blockchain was proposed to provide secure charging services for electric vehicles [20]. An IoT E-business model was proposed in [21], where data of transactions were recorded on the blockchain. However, only a few studies have applied the blockchain to the scenario of spectrum sharing in wireless

communication networks. A Spectrum Sensing as a Service (SPASS) architecture was proposed to realize the payment transferring of outsourced spectrum sensing services through the smart contract [23]. The feasibility of applying the smart contract-enabled blockchain technology to the spectrum sharing was analyzed in [24] from five aspects, including the decentralization, the transparency, the irreversibility, the availability and the security. In [26], considering that some home or business users could receive a certain revenue by providing spectrum sensing services for nodes in SU networks, a smart contract was designed to implement Service Level Agreements (SLAs) between mobile OPs and community service providers. However, the efficient spectrum sharing based on the blockchain is still an open issue due to the high block verification delay and low throughput.

### III. SYSTEM FRAMEWORK

In this section, the system framework based on the permissioned blockchain for the spectrum sharing among multi-OPs in wireless communication networks is shown in Figure 1. The MBS and small cell base stations (SBS) of multi-OPs coexist in the specific region, where each OP includes a MBS covering a number of SBSs. The MBS and SBSs of each OP analyze the spectrum usage and user locations reported by user terminals in the specific region. The MBS and SBSs of each OP report analysis results to the corresponding servicer, i.e., the Operation Administration and Maintenance (OAM) servicer, which is responsible for the spectrum management among associated MBS and SBSs. Based on the analysis results collected from the MBS and SBSs, the OAM servicer of each OP can reasonably predict the spectrum demand during the period of  $[t_b, t_e]$ .

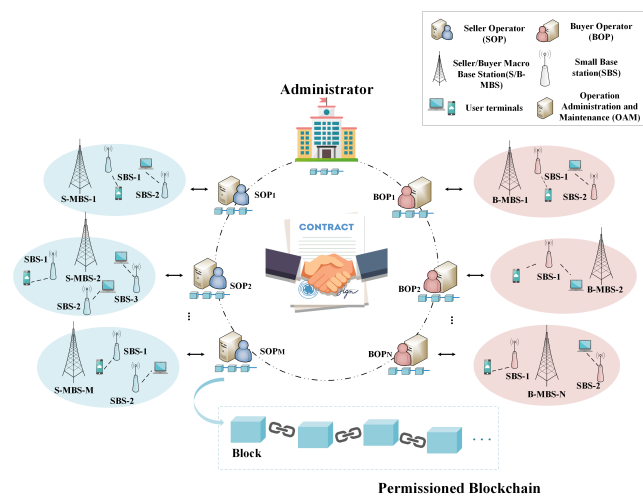


FIGURE 1. System framework.

#### A. MAIN ENTITIES OF THE SPECTRUM SHARING NETWORK

A spectrum trading market can be established to satisfy different spectrum demands of multi-OPs. The MOSS smart

contract is designed to realize the spectrum trading and allocation among multi-OPs. The proposed permissioned blockchain trust framework consists of three types of entities for the spectrum sharing in multi-OPs wireless communication networks: seller OPs group, buyer OPs group and the administrator. These three entities are connected and communicated through the constructed blockchain networks. The main entities of the proposed Spectrum Sharing Framework are listed as follows.

**Seller OPs group:** The seller OPs group is constituted by  $\mathcal{M}(\mathcal{M}=\{1, 2, \dots, M\})$  OPs. The  $m$ -th OP decides to sell  $B_m$  ( $B_m \leq B_m^{tot}, m \in \mathcal{M}$ ) bandwidth, where  $B_m^{tot}$  is the total bandwidth owned by the  $m$ -th OP. In order to guarantee the quality of service (QoS) of service users associated with the  $m$ -th OP during the period of  $[t_b, t_e]$ , the  $m$ -th OP retains  $B_m^{left}$  bandwidth to serve associated users.  $B_m^{left}$  meets the following requirement:

$$B_m^{left} = B_m^{tot} - B_m \geq B_m^{req}, \quad \forall m \in \mathcal{M}, \quad (1)$$

where  $B_m^{req}$  is the bandwidth required by the  $m$ -th OP to provide the specific QoS for the associated users,  $p_m$  is the price per unit bandwidth bid by the  $m$ -th OP.

**Buyer OPs group:** The buyer OPs group is constituted by  $\mathcal{N}(\mathcal{N} = \{1, 2, \dots, N\})$  OPs. The  $n$ -th OP decides to buy  $W_n$  bandwidth based on the prediction of spectrum demand during the period of  $[t_b, t_e]$ , and the bid price per unit bandwidth is set as  $c_n$ .

**Administrator:** The administrator is responsible for supervising the implementation process of spectrum sharing among multi-OPs. The main duties of the administrator are listed as follows:

- 1) The administrator is responsible for the certificate authentication and the off-chain supervision of OPs. Each OP need to be authenticated the identity before acting as a node in the blockchain. Once supervising the malicious behaviour of OPs, the administrator can record it on the blockchain and punish malicious OPs according to the pre-defined rules in the smart contract.
- 2) Considering the fact that the deployment of smart contract needs to cost large gas and selfish OPs are unwilling to pay for it, the administrator can serve as the deployer of the MOSS smart contract and the caller of some specified functions. What to mention is that the administrator just acts as the deployer and caller without participating in the process of spectrum allocation as a spectrum broker.

In the proposed framework, the MOSS smart contract is deployed at time  $t_0$  by the administrator. Simultaneously, the bidding time is initialized as  $t_{bid}$  and the free-trading time is initialized as  $t_{free}$  when the MOSS smart contract is deployed. The free-trading market is triggered at time  $t_1(t_1 > (t_0 + t_{bid}))$  by the administrator. The valid bidding duration is  $[t_0, t_0 + t_{bid}]$  ( $(t_0 + t_{bid}) < t_b$ ), and the valid free-trading duration is  $[t_1, t_1 + t_{free}]$  ( $(t_1 + t_{free}) < t_b$ ). The more details will be introduced in Section IV.

## B. PERMISSIONED BLOCKCHAIN-ENABLED SPECTRUM SHARING OPERATIONS

To guarantee the trusted spectrum sharing among multi-OPs in 5G wireless communication networks, a permissioned blockchain-enabled spectrum sharing framework is established. The general spectrum sharing operations for entities in the permissioned blockchain are described as follows.

- 1) Initializing the system: the permissioned blockchain is only available to the authenticated entities [27]. Before joining in the permissioned blockchain, the OP  $i$  ( $i \in \mathcal{M} \cup \mathcal{N}$ ) needs to register with the administrator to be authorized with the identity  $ID_i$ , the public key  $PK_i$ , the private key  $SK_i$ , the legitimate certificate  $Cert_i$  and the wallet address  $WA_i$ . The mapping list  $\{ID_i, PK_i, Cert_i, WA_i\}$  can be sent as a transaction by the OP  $i$  to adjacent authorized OPs. All the transaction records are stored in the memory pool of OAM of authorized OPs. The OAM of each OP is considered to have a certain storage and computing power. Based on the stored certificate information of each registered OP, the authorized OPs can verify the identity of transaction sender.
- 2) Sending transactions: after initialization, the administrator can deploy the MOSS smart contract to start the spectrum sharing among multi-OPs. Multi-OPs can choose to become the seller OPs or buyer OPs based on the spectrum usage state. Seller OPs and buyer OPs can trade the spectrum by calling specific functions defined in the MOSS smart contract. More details about the MOSS smart contract will be given in the section IV.
- 3) Performing the consensus process: the PBFT consensus algorithm is used to reach a consensus about the spectrum allocation result among registered OPs. Multi-OPs who want to participate in the consensus process can become a replica in the PBFT algorithm, and one of OPs can be selected as the primary. A set of transactions are verified and packaged into the block by the primary. The primary broadcasts the block to all replicas for auditing. The audited results of each replica are announced among all replicas in the permissioned blockchain. By comparing the received audited results, each replica can send the feedback information to the primary. After reaching a consensus, the block data will be broadcasted by the primary to each entity in the permissioned blockchain and a newly verified block can be added to the blockchain.

Underloaded OPs who may have the idle spectrum during the period of  $[t_b, t_e]$  can join the seller OPs group and share their idle spectrum with a certain price. As a consequence, the revenue and spectrum utilization of underloaded OPs can be improved. Overloaded OPs, whose licensed spectrum is unable to meet needs of all users during the period of  $[t_b, t_e]$ , can join the buyer OPs group to purchase the spectrum from the seller OPs group. With the spectrum purchased from seller OPs, the QoS of users served by overloaded OPs can be improved.

Both the spectrum trading and payment transferring between seller OPs and buyer OPs are executed by the designed MOSS smart contract. Each entity has a unique account. Transactions are recorded in the constructed permissioned blockchain. During the period of  $[t_0, t_0 + t_{bid}]$ ,  $M$  seller OPs and  $N$  buyer OPs independently register and set the bid through the MOSS smart contract. As the registration period ends, the administrator invokes the MOSS smart contract to perform the spectrum auction among registered seller and buyer OPs. If there exist unsuccessful matches between buyer OPs and seller OPs, the administrator can invoke the MOSS smart contract to open the free-trading market. Unsuccessfully matched OPs can choose whether to enter the free-trading market during the period of  $[t_1, t_1 + t_{free}]$ , and adjust their bids based on matched results of the spectrum auction stage. Finally, all OPs can invoke the MOSS smart contract to take back the rest of their deposit. A round of spectrum sharing among multi-OPs during the period of  $[t_b, t_e]$  is finished.

## IV. SMART CONTRACT DESIGN

In this section, the flow of designed MOSS smart contract is introduced. The explanation of parameters are shown in Table 1. The smart contract is mainly consisted of functions and events [28]. The functions are the executable code to implement the specified functionality, and the event can be used to notify the authenticated nodes of the state change in the blockchain [29].

TABLE 1. Explanation of parameters.

Notations	Descriptions
$t_0$	Time to deploy the MOSS
$t_{bid}$	Time of bidding registration
$t_{free}$	Time of free-trading market
$t_1$	Time to begin the free-trading market
$B_m$	The bandwidth sold by the $m$ -th op
$p_m$	Price per unit bandwidth of the $m$ -th op
$W_n$	The bandwidth bought by the $n$ -th op
$c_n$	Price per unit bandwidth of the $n$ -th op
$asks$	The registration set of seller OPs
$bids$	The registration set of buyer OPs
deposit	The account balance of OP
$addr_m^S$	address of the $m$ -th op
$addr_n^B$	address of the $n$ -th op
$msg.sender$	The sending address of transaction
$\_addr$	The address of OP in the free-trading market
$\_price$	The latest price of OP in the free-trading market
$\_bandwidth$	The latest bandwidth of OP in the free-trading market
$role$	The seller/buyer OP

In the MOSS smart contract, the multi-OPs spectrum sharing process is divided into three stages: the registration of OPs, the spectrum trading and the payment clearing. Functions defined in the MOSS smart contract should be agreed by all OPs before the MOSS smart contract is deployed in the permissioned blockchain. The administrator sends a transaction to deploy the MOSS smart contract at time  $t_0$ . The address of MOSS smart contract is known by all OPs

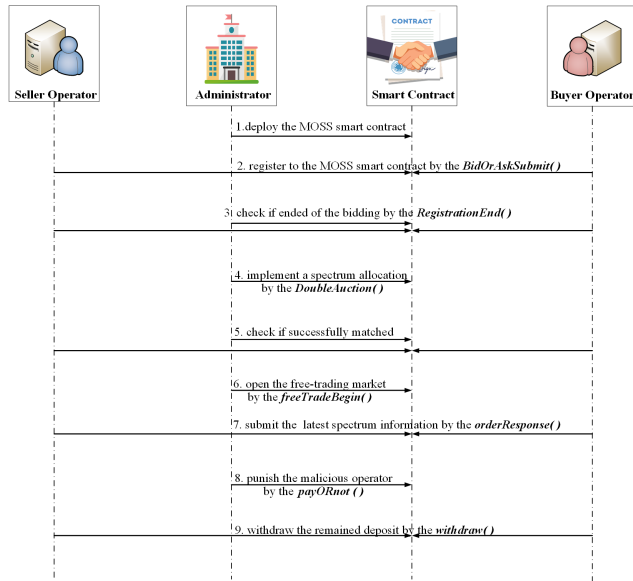


FIGURE 2. Flow of the designed MOSS smart contract.

once this transaction is packaged into the valid block. The timestamp in the blockchain can be used in the MOSS smart contract to judge the timeliness of transactions. The function call that is invalid in the current period will not be packaged into the block by miners. The detailed flow of MOSS smart contract is shown in Figure 2 (the source code of MOSS smart contract can be seen in the github at [https://github.com/cherry1124/MOSS\\_CONTRACT](https://github.com/cherry1124/MOSS_CONTRACT)) and is described as follows.

### A. THE REGISTRATION OF OPs

In the registration stage of OPs, the *BidOrAskSubmit()* function is designed to submit the bid of multi-OPs and the *AuctionEnd()* function is designed to judge whether the registration process is ended. The specific functionalities of *BidOrAskSubmit()* and *AuctionEnd()* are described as follows.

*BidOrAskSubmit()*: During the period of  $[t_0, t_0 + t_{bid}]$ , OPs call the *BidOrAskSubmit()* when they want to participate in the spectrum sharing and declare whether they are buyer OPs or seller OPs. The bandwidth and price per unit bandwidth they want to buy or sell also need to be given. The *BidOrAskSubmit()* function is set as payable, which represents that the *BidOrAskSubmit()* have the functionality of ether transferring. All OPs must deposit more than 1 ether into the address of MOSS smart contract when the *BidOrAskSubmit()* is called. The ether is the common digital currency in the Ethereum. The requirement of deposit can protect the spectrum trading process against interferences from some malicious nodes and be used for the payment transferring of spectrum trading. The remaining deposit can be returned back to the account of corresponding OP. The *LogRegisterOp* event is defined in the *BidOrAskSubmit()*. If OPs successfully register in the MOSS smart contract and

submit their bid, the *LogRegisterOp* event will be triggered. All blockchain nodes listening for this event will know the bid information of registered OPs, from which unregistered OPs can dynamically adjust their demands and bids. The bid information is sent as a transaction via the anonymous address of each OP. Therefore, the real identity of OPs will not be revealed and the private information of OPs will be protected.

*RegistrationEnd()*: Multi-OPs can call the *RegistrationEnd()* function at any time to determine whether the registration stage is ended. The timestamp of transactions sent by OPs is represented by *now*. If  $now > (t_0 + t_{bid})$ , the registration stage is already ended. Therefore OPs cannot call the *BidOrAskSubmit()* function to submit the bid and cannot participate in this round of spectrum sharing.

### B. THE SPECTRUM TRADING

The process of spectrum trading is divided into two sub-stages. The first sub-stage is the spectrum auction among multi-OPs, which mainly includes four functions: the *sortAskByIncrease()*, the *sortBidByDecrease()*, the *DoubleAuction()* and the *doubleAuctionFinish()*. The second sub-stage is mainly designed for OPs who fail to match in the stage of spectrum auction and provides a free-trading market for unsuccessfully matched OPs. The second sub-stage includes four functions: the *freeTradeBegin()*, the *orderResponse()*, the *deleteOrder()*, and the *MarketEnd()*. The specific functionalities of functions contained in two sub-stages are described as follows.

*sortAskByIncrease()*: As the registration stage of OPs ended, the administrator calls the *sortAskByIncrease()* in the MOSS smart contract to sort M seller OPs by the price in ascending order.

*sortBidByDecrease()*: The administrator also calls the *sortBidByDecrease()* to sort N buyer OPs by the price in descending order.

*DoubleAuction()*: The administrator calls the *DoubleAuction()* to auction the spectrum for all registered OPs after the sorting is completed. The process of *DoubleAuction()* is shown in Algorithm 1. The consensus nodes who packages the specific transaction into the block performs the corresponding operations:

- 1) registered buyer OPs and seller OPs are matched based on the spectrum auction algorithm defined in the *DoubleAuction()*. The successfully matched price per unit bandwidth is set as the average price of the matched buyer OP and seller OP.
- 2) Second, based on the matched price and bandwidth quantity, the expense of buyer OPs and the income of seller OPs are calculated.
- 3) Third, the spectrum auction is finished until the bidding queue have been fully matched or the lowest seller price is higher than the highest buyer price.

The variable *doubleAuctionFinish* defined in the MOSS smart contract represents the state of spectrum auction. When the value of *doubleAuctionFinish* changes from false to true, all OPs and the administrator who inquire the

**Algorithm 1** Spectrum Auction

---

**Input:**  $asks = \{(p_1, B_1, addr_1^S), \dots, (p_M, B_M, addr_M^S)\}$   
 $// (p_1 \leq \dots \leq p_M)$   
 $bids = \{(c_1, W_1, addr_1^B), \dots, (c_N, W_N, addr_N^B)\}$   
 $// (c_1 \geq \dots \geq c_N)$   $doubleAuctionFinish = false$

**Output:** Successfully matched result

- 1  $bidsLength = bids.length;$
- 2  $asksLength = asks.length;$
- 3  $doubleAuctionFinish = false;$
- 4 **while**
- $bidsLength \neq 0 \ \& \ asksLength \neq 0 \ \& \ (c_1 \geq p_1)$  **do**
- 5      $dealPrice = (p_1 + c_1) / 2;$
- 6      $dealAmount = \min\{B_1, W_1\};$
- 7      $totalMoney = dealPrice * dealAmount;$
- 8      $deposit[addr_1^B] = deposit[addr_1^B] - totalMoney;$
- 9      $deposit[addr_1^S] = deposit[addr_1^S] + totalMoney;$
- 10     $(p_1, B_1) \leftarrow (p_1, B_1 - dealAmount);$
- 11     $(c_1, W_1) \leftarrow (c_1, W_1 - dealAmount);$
- 12    **if**  $(W_1 == 0)$  **then**
- 13     **for**  $(c_n, W_n, addr_n^B) \in bids$  **do**
- 14     |  $(c_{n+1}, W_{n+1}, addr_{n+1}^B) \leftarrow (c_n, W_n, addr_n^B)$
- 15     **end**
- 16    **else**
- 17     **if**  $(B_1 == 0)$  **then**
- 18     | **for**  $(p_m, B_m, addr_m^S) \in asks$  **do**
- 19     | |  $(p_{m+1}, B_{m+1}, addr_{m+1}^S) \leftarrow (p_m, B_m, addr_m^S)$
- 20     | **end**
- 21     **end**
- 22    **end**
- 23 **end**
- 24  $doubleAuctionFinish = true$

---

$doubleAuctionFinish$  can know that the spectrum allocation has already completed. The *LogDealRecord* event is defined in the MOSS smart contract. When buyer OPs and seller OPs have a successful match, the *LogDealRecord* will be triggered. All nodes listening for the *LogDealRecord* will know the successful match information. The repeated auction of spectrum and the mutual interferences of spectrum usage are avoided based on the proposed spectrum allocation scheme. The openness of spectrum sharing can be guaranteed in the proposed solution.

*freeTradeBegin()*: If there exist unsuccessful matched OPs in the *DoubleAuction()*, the administrator will call the *freeTradeBegin()* at time  $t_1$  to open the free-trading market. The valid period of free-trading market is  $[t_1, t_1 + t_{free}]$ .

*orderResponse()*: During the period of  $[t_1, t_1 + t_{free}]$ , OPs who fail to match in the *DoubleAuction()* can choose whether to enter the free-trading market. Based on the matched information in the spectrum auction stage, seller OPs who decide to enter the free-trading market can call the *orderResponse()*

to resubmit the adjusted price and the bandwidth demand. Once the latest bid information is submitted successfully, the *LogFreeMarketOrder* event is triggered to declare the newly updated price information of OPs. Buyer OPs can call the *orderResponse()* to purchase the desired spectrum based on the latest spectrum information and reach a successful match. All the updated bid information and successful match result will be recorded in the permissioned blockchain by consensus nodes. The flow of *orderResponse()* is shown in Algorithm 2.

**Algorithm 2** Free-Trading Market

---

**Input:**  $\_addr, \_price, \_bandwidth$

**Output:** Successfully matched result

- 1 **if** Buyer OPs purchase the spectrum **then**
- 2 |  $new\_price = \_price;$
- 3 |  $new\_band = \_bandwidth;$
- 4 **end**
- 5 **if** OPs resubmit the adjusted bids **then**
- 6 | **if**  $(\_bandwidth \leq \_addr.bandwidth)$  **then**
- 7 | |  $totalMoney = \_price * \_bandwidth;$
- 8 | |  $deposit[msg.sender] \leftarrow$
- 9 | |  $deposit[msg.sender] - totalMoney;$
- 10 | |  $deposit[\_addr] \leftarrow$
- 11 | |  $deposit[\_addr] + totalMoney;$
- 12 | |  $msg.sender.bandwidth = 0;$
- 13 | |  $\_addr.bandwidth =$
- 14 | |  $\_addr.bandwidth - \_bandwidth;$
- 15 | **else**
- 16 | | **if**  $(\_bandwidth > \_addr.bandwidth)$  **then**
- 17 | | |  $totalMoney = \_price * \_addr.bandwidth;$
- 18 | | |  $deposit[msg.sender] \leftarrow$
- 19 | | |  $deposit[msg.sender] - totalMoney;$
- 20 | | |  $deposit[\_addr] \leftarrow$
- 21 | | |  $deposit[\_addr] + totalMoney;$
- 22 | | |  $msg.sender.bandwidth \leftarrow$
- 23 | | |  $msg.sender.bandwidth - \_bandwidth;$
- 24 | | |  $\_addr.bandwidth = 0;$
- 25 | | **end**
- 26 **end**
- 27 **end**

---

*deleteOrder()*: If the OP fails to match in the *DoubleAuction()* and refuses to enter the free-trading market, the OP can call the *deleteOrder()* to clear the bid information and exit the spectrum sharing market.

*MarketEnd()*: Multi-OPs and the administrator can call the *MarketEnd()* to determine whether the free-trading market is ended. If  $now > (t_1 + t_{free})$ , the OP cannot call the *orderResponse()* since the free-trading market has already ended. When the free-trading market is ended, the whole spectrum sharing process is finished. If there exist OPs who still want to join the spectrum sharing, they can only wait for the next new round of spectrum sharing.

### C. THE PAYMENT CLEARING

In order to implement the punishment of malicious OPs and the payment transferring of registered OPs, three functions including the *payORnot* ( ), the *increaseFunds*( ) and the *withdraw*( ) are designed in the MOSS smart contract. The specific functionalities of *payORnot* ( ), *increaseFunds*( ) and *withdraw*( ) are described as follows.

*payORnot*( ): A function modifier *ownerOnly* is set in the *payORnot* ( ), which regulates that the *payORnot* ( ) can only be called by the administrator. Other blockchain nodes do not have the permission to call the *payORnot* ( ). During the period of  $[t_b, t_e]$ , the administrator is responsible for ensuring that the exchange of spectrum usage right is executed off-chain correctly. The bool variable *executeORnot* is defined in the MOSS smart contract to represent whether the corresponding OP correctly exchanges the spectrum usage right. If the OP does not exchange the spectrum usage right based on matched results of the MOSS smart contract, the administrator can call the *payORnot* ( ) to set the *executeORnot* value of corresponding OP as false. Thus malicious OPs are prevented from withdrawing their remaining deposit. As for honest OPs, the administrator calls the *payORnot* ( ) to set the *executeORnot* of honest OPs as true and honest OPs can successfully withdraw their remaining deposit.

*increaseFunds*( ): The *increaseFunds*( ) is defined as payable for multi-OPs to increase their deposit in the MOSS smart contract. The *increaseFunds*( ) is called by OPs when the remaining deposit is not enough to be transferred to other OPs. Thus the payment transferring of spectrum trading can be executed smoothly.

*withdraw*( ): All OPs participating in the spectrum sharing will call the *withdraw*( ) to get back the remaining deposit at the end time of  $[t_b, t_e]$ . When OPs whose value of the *executeORnot* are false call the *withdraw*( ) to get back the remaining deposit, the consensus nodes will not package this transaction. Honest Ops whose value of the *executeORnot* are true can smoothly take back the remaining deposit by calling the *withdraw*( ).

The proposed MOSS smart contract can resolve the problem with the existence of malicious OPs in the spectrum sharing. The administrator can punish malicious OPs through the MOSS smart contract, making them unable to take back the deposit. A friendly and secure spectrum sharing market can be built by prompting all OPs honestly exchange the spectrum usage right.

### V. SYSTEM IMPLEMENTATION AND ANALYSIS

In this section, the designed MOSS smart contract is implemented and tested in the Remix IDE found at <http://remix.ethereum.org>. The Remix is a browser-based IDE for developing smart contracts without the need to install the Solidity environment, which is an object-oriented, high-level language for implementing smart contracts. The performance analysis of the proposed solution are also given in the following.

### A. TESTING OF THE MOSS SMART CONTRACT

In order to test and estimate the cost of MOSS smart contract visually, the designed MOSS smart contract is compiled in the Remix IDE and deployed in the Ganache, which is used for the test case. Nodes can send transactions to deploy the MOSS smart contract and call functions through their unique addresses. The gas price and gas limit can be independently set by OPs when transactions are sent.

Three seller OPs and three buyer OPs are considered to test the feasibility of the MOSS smart contract. The addresses and spectrum demands of all entities are shown in Table 2. Based on accounts provided by the Ganache, the MOSS smart contract is tested and the gas consumption of each function in the MOSS smart contract is estimated. The Ether consumption can be calculated as

$$\text{Ether\_consumption} = \text{Gas\_cost} * \text{Gas\_price}. \quad (2)$$

TABLE 2. Accounts and spectrum bid information of each entity.

Role	Account Address	Amount(MHZ)	Price(Gwei/MHZ)
administrator	0xd769 ... e4AC	-	-
seller OP <sub>1</sub>	0x36d2 ... E1fE	20	2000000
seller OP <sub>2</sub>	0x5E6E ... f1A0	10	1600000
seller OP <sub>3</sub>	0x0004 ... 561d	15	2400000
buyer OP <sub>4</sub>	0x5B84 ... 2631	10	1500000
buyer OP <sub>5</sub>	0x8c66 ... 1155	12	2500000
buyer OP <sub>6</sub>	0x9508 ... bcaa	8	1800000

Based on the current recommended gas price found at <https://ethgasstation.info/>, the gas price is set as 4.3 Gwei, where wei is the minimum unit of ether and  $1 \text{ ether} = 10^9 \text{ Gwei} = 10^{18} \text{ wei}$ . Considering that the gas consumption is related to different operations defined in the functions, all the gas consumption corresponded to the possible function execution results are shown in Table 3. The corresponding Ether cost is also calculated and given in Table 3. From Table 3, the Ether cost of deploying the MOSS smart contract is 0.0204989 *eth*. Among all functions called by the administrator in the MOSS smart contract, the function which costs the most amount of Ether is *DoubleAuction*( ) with the cost of 0.001583935 *eth*. The Ether cost of remaining functions are modest.

TABLE 3. Gas cost of functions called by the Administrator.

Function	Transaction Gas	Gas cost in Ether
MOSS smart contract	4767204 gas	$2.04989 \times 10^{-2}$
<i>RegistrationEnd</i> ( )	21799 gas	$9.37357 \times 10^{-5}$
<i>sortAskByIncrease</i> ( )	70696 gas	$3.0399 \times 10^{-4}$
<i>sortBidByDecrease</i> ( )	116557 gas	$5.01195 \times 10^{-4}$
<i>DoubleAuction</i> ( )	368357 gas	$1.583935 \times 10^{-3}$
<i>freeTradeBegin</i> ( )	42413 gas	$1.82375 \times 10^{-4}$
<i>MarketEnd</i> ( )	21776 gas	$9.35938 \times 10^{-5}$
<i>payORnot</i> ( )	29018 gas	$1.24777 \times 10^{-4}$
<i>changeOwner</i> ( )	28811 gas	$1.23887 \times 10^{-4}$
<i>selfDestruct</i> ( )	13495 gas	$5.8028 \times 10^{-5}$

The gas consumption of all functions called by OPs and the cost of Ether are shown in Table 4. From Table 4, it is

TABLE 4. Gas cost of functions called by OPs.

Function	Transaction Gas	Gas cost in Ether
<i>BidOrAskSubmit()</i>	216416 gas	$1.00362 \times 10^{-3}$
<i>deleteOrder()</i>	21229 gas	$9.12847 \times 10^{-5}$
<i>orderResponse()</i>	24277 gas when buyer OPs call this function; 35085 gas when seller OPs want to change the bid	$1.0439 \times 10^{-4} \sim 1.5086 \times 10^{-4}$
<i>withdraw()</i>	22188 gas	$9.5408 \times 10^{-5}$
<i>increaseFunds()</i>	26757 gas	$1.15055 \times 10^{-4}$

obvious that only the *BidOrAskSubmit()*, the cost of which is 0.001 eth, consumes the most gas among all functions called by OPs in the MOSS smart contract. And the overall cost of remaining functions called by OPs are modest.

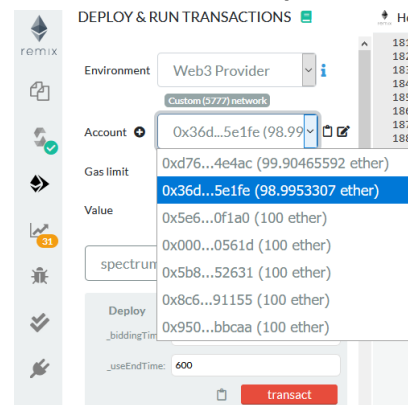
From Table 3 and Table 4, the gas cost called by OPs are almost negligible compared to the cost of functions called by the administrator. OPs can trade the spectrum with the modest cost, which greatly motivates OPs to participate in the proposed spectrum sharing market and improves the spectrum utilization. Different from the spectrum broker that exists as a third party in previous studies, the administrator does not directly participate in the spectrum allocation among multi-OPs in our proposed solution. The spectrum allocation among multi-OPs is implemented based on predefined functions in the MOSS smart contract. The administrator only acts as a deployer and caller of functions in the MOSS smart contract, and supervises the entire spectrum sharing process off-chain.

The *LogRegisterOp* event is triggered when OPs call the *BidOrAskSubmit()*. Figure 3(a) shows the transaction log when the seller OP<sub>1</sub> calls the *BidOrAskSubmit()* to submit the bid information. From the log field shown in Figure 3(a), the address of OP<sub>1</sub> is  $0 \times 36d\dots E1Fe$ , the amount of bandwidth that the OP<sub>1</sub> wants to sell is 20MHz in the simulation and the price per unit bandwidth is set as 2000000 *Gwei*. All nodes in the blockchain that listen to the *LogRegisterOp* will receive the latest registration information of OP<sub>1</sub>. Other OPs can reasonably adjust their own bids based on the bid information of OP<sub>1</sub>. Meanwhile, OPs can set their own price in the next round of spectrum sharing by learning the bid information of other OPs in each round. The probability of successful match in the spectrum auction process will increase. The account balance of OP<sub>1</sub> is almost 98.99 eth shown in Figure 3(b). The OP<sub>1</sub> is required to deposit 1 eth in the MOSS smart contract when the *BidOrAskSubmit()* is called. The gap between the account balance of OP<sub>1</sub> and 99 eth is the transaction fee when the transaction is sent to call the *BidOrAskSubmit()*.

Figure 4 shows the transaction log when the administrator calls the *DoubleAuction()*. The predefined spectrum auction algorithm in the *DoubleAuction()* is executed by consensus nodes. If there exist successfully matches among multi-OPs in the spectrum auction, the *LogDealRecord* is triggered. It can be seen from the log field in Figure 4 that the seller OP<sub>2</sub> and the buyer OP<sub>5</sub> have successfully traded 10 MHz of



(a) Transaction log



(b) Account balance

FIGURE 3. Calling the *BidOrAskSubmit()* by OP<sub>1</sub>.



FIGURE 4. Transaction log of calling the *DoubleAuction()* by the Administrator.

spectrum at a price of 2050000 *Gwei*. Also, the seller OP<sub>1</sub> and the buyer OP<sub>5</sub> have successfully traded 2 MHz of spectrum at a price of 2250000 *Gwei*. The remaining OPs fail to match in the *DoubleAuction()* stage.

After the spectrum auction, the *freeTradeBegin()* is called by the administrator to open the spectrum free-trading market among unsuccessfully matched OPs. In the simulation, the seller OP<sub>1</sub> is assumed to enter the free-trading market and modify the bid price to 1800000 *Gwei* by calling the *orderResponse()*. The remaining unsuccessfully matched buyer



```

logs
[
  {
    "from": "0x36d2668f1c2136e19645a7a32eb1b1a2081fe",
    "topic": "0x934c3ef7037c90e0b2aaa896e43cbe01c4072a5796f84e3c4947243131",
    "event": "LogFreeMarketOrder",
    "args": {
      "op": "0x36d2668f1c2136e19645a7a32eb1b1a2081fe",
      "role": "1",
      "price": "1800000000000000",
      "amount": "8",
      "length": 4
    }
  }
]
    
```

FIGURE 5. Transaction log of calling the *orderResponse()* by the buyer OP<sub>6</sub>.

```

transact to spectrumSharing.withdraw pending ...
transact to spectrumSharing.withdraw errored: VM Exception while processing transaction: revert Invalid op
    
```

FIGURE 6. Transaction log of calling the *withdraw()* by the seller OP<sub>2</sub>.

OPs can decide whether to participate in the free-trading market based on the latest bid information of OP<sub>1</sub>. If the buyer OP<sub>6</sub> decides to purchase the spectrum after listening to the modified bid information of OP<sub>1</sub>, the buyer OP<sub>6</sub> can call the *orderResponse()* to declare the desired spectrum information and reach a successful match. It can be seen from the transaction log field in Figure 5 that the buyer OP<sub>6</sub> and the seller OP<sub>1</sub> have successfully traded 8 MHz spectrum at a price of 1800000 *Gwei*.

When the free-trading market is ended, the entire spectrum sharing process is completed. Each OP can call the *withdraw()* to withdraw the remaining deposit in the MOSS smart contract. The seller OP<sub>2</sub> is assumed to be a malicious OP in the simulation. After successfully matched with the buyer OP<sub>5</sub> in the previous spectrum auction, the traded spectrum usage right of OP<sub>2</sub> is not correctly delivered to the corresponding buyer OP<sub>5</sub> within the period of  $[t_b, t_e]$ . Once supervising the off-chain malicious behaviour of OP<sub>2</sub>, the administrator can call the *payORnot()* to set the *executeORnot* of OP<sub>2</sub> as false and freeze the account balance of OP<sub>2</sub>. If the *withdraw()* function is called to withdraw the deposit by the seller OP<sub>2</sub>, the transaction is rolled back and declared as “Invalid op” shown in Figure 6. Other honest OPs can successfully withdraw their deposits by calling the *withdraw()*. It can be seen from Figure 7(a) that the transaction log field of honest OP<sub>1</sub> is shown as “successful to withdraw.” The account balance of OP<sub>1</sub> is more than 100 *eth* shown in Figure 7(b), which indicates that the seller OP<sub>1</sub> can withdraw its 1 *eth* deposit and earn the extra revenue from selling the spectrum usage right.

**B. PERFORMANCE ANALYSIS**

1) SECURITY ANALYSIS

In traditional multi-OPs spectrum sharing solutions, the spectrum broker is introduced to allocate and manage spectrum resources. The problem of privacy disclosure exists when multi-OPs interact with the spectrum broker. Besides, the data sent by multi-OPs to the trustless spectrum broker may be maliciously tampered with. In this paper, the distributed chained-block structure of blockchain technology is used to

```

transaction hash 0x5890309faf8259387190ebaaclea91199e12f685f2e490982b8bfc0d2f133
from 0x36d2668f1c2136e19645a7a32eb1b1a2081fe
to spectrumSharing.withdraw(0) 0x2debdbf6f6a5e0856cb22a1784cf0e75c5b84e
gas 37740 gas
transaction cost 22188 gas
hash 0x5890309faf8259387190ebaaclea91199e12f685f2e490982b8bfc0d2f133
input 0x3c...f460b
decoded input {}
decoded output {}
logs [
  {
    "from": "0x2debdbf6f6a5e0856cb22a1784cf0e75c5b84e",
    "topic": "0x083704268f1a1dca5c3882aa8e6787af6b612ad944a76cfed4ca90026cfab5",
    "event": "LogNotice",
    "args": {
      "op": "Successful to withdraw",
      "msg": "Successful to withdraw",
      "length": 1
    }
  }
]
value 0 wei
    
```

(a) Transaction log

(b) Account balance

FIGURE 7. Calling the *withdraw()* by OP<sub>1</sub>.

realize the spectrum sharing among multi-OPs. All transactions and spectrum usage status are recorded in a data structure of merkle tree in the blockchain. All transactions need to be signed with the private key of transaction sender and verified by the public key of transaction receiver. The private key and the public key of nodes are generated by the asymmetric encryption algorithm. Each consensus node in the blockchain locally maintains a distributed ledger to record all transactions broadcasted in the blockchain. If there exist malicious OPs who want to tamper with the transaction information, honest OPs who account for a large proportion will not modify their own local ledger. The integrity and security of data can be ensured by the proposed solution. Besides, the permissioned blockchain among multi-OPs is constructed in this paper. Different from the public blockchain, the private data of OPs is invisible to nodes outside the permissioned blockchain. Moreover, through the certificate authentication (CA) of permissioned blockchain, strict identity management is conducted for all OPs participating in the spectrum sharing. The privacy preserving of multi-OPs in the spectrum sharing process can be realized in the proposed solution.

2) RELIABILITY ANALYSIS

Based on the decentralization characteristic of blockchain, each OP in the permissioned blockchain maintains a local ledger in a distributed structure. When one of OPs fails, other OPs can still normally operate. The single point of failure is

avoided in the proposed solution, which is more reliable than traditional centralized spectrum sharing solutions. Besides, the MOSS smart contract is designed to implement the spectrum allocation among multi-OPs, eliminating the existence of centralized spectrum broker in traditional methods. Before starting the spectrum sharing, multi-OPs jointly design the logic of spectrum allocation scheme in the MOSS smart contract. The entire spectrum sharing process is automatically executed based on the agreed functions in the MOSS smart contract, which ensures the fairness and openness of spectrum sharing in multi-OPs wireless communication networks.

**TABLE 5. Comparison of proposed framework with existing researches.**

Characteristic	[7]	[8]	[15]	[25]	Proposed solution
Decentralized	×	✓	✓	✓	✓
Privacy	×	×	×	✓	✓
Traceable	×	×	✓	✓	✓
MOSS	×	×	×	×	✓
fairness	✓	✓	✓	✓	✓
Openness	×	×	×	✓	✓

### 3) COMPARISON ANALYSIS

In order to have an intuitive analysis, our proposed framework is compared to other three existing researches of the spectrum sharing among multi-OPs. The comparison results are shown in Table 5. The analysis results demonstrate that the privacy, openness and fairness of the proposed solution are better than other three spectrum sharing solutions.

## VI. CONCLUSION

In this paper, a permissioned blockchain trust framework is proposed for the spectrum sharing in multi-OPs wireless communication networks. The MOSS smart contract is designed to automatically perform the spectrum trading and payment transferring among multi-OPs. A fair spectrum sharing market among multi-OPs can be realized without the existence of third party. The gas consumption of each function defined in the MOSS smart contract is tested. The security and the reliability analysis of the proposed solution are evaluated. The simulation results and the performance analysis demonstrate that the privacy, openness and fairness of the proposed solution are better than traditional spectrum allocation solutions.

In the future work, the incentive mechanism will be considered to incentive OPs to share their spectrum, which can further improve the spectrum efficiency. Besides, designing an efficient auction algorithm will be considered to further reduce the gas cost of calling functions in the MOSS smart contract.

## REFERENCES

- [1] P. Kolodzy and I. Avoidance, "Spectrum policy task force," Federal Commun. Commission, Washington, DC, USA, Tech. Rep. 02-135, 2002.
- [2] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," Cisco Vis. Netw. Index, San Jose, CA, USA, White Paper 11, 2019, pp. 1–37.
- [3] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, 1999.
- [4] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [5] K. Koufos, *Deliverable D5. 4 Future Spectrum System Concept*. [Online]. Available: [https://www.metis2020.com/wp-content/uploads/deliverables/METIS\\_D5.4\\_v1.pdf](https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D5.4_v1.pdf)
- [6] S. Pandit and G. Singh, "An overview of spectrum sharing techniques in cognitive radio communication system," *Wireless Netw.*, vol. 23, no. 2, pp. 497–518, Feb. 2017.
- [7] Q. Deng, Z. Li, J. Chen, F. Zeng, H.-M. Wang, L. Zhou, and Y.-J. Choi, "Dynamic spectrum sharing for hybrid access in OFDMA-based cognitive femtocell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10830–10840, Nov. 2018.
- [8] T. Sanganpuak, S. Guruacharya, N. Rajatheva, M. Bennis, and M. Latva-Aho, "Multi-operator spectrum sharing for small cell networks: A matching game perspective," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3761–3774, Jun. 2017.
- [9] L. Gao, X. Wang, Y. Xu, and Q. Zhang, "Spectrum trading in cognitive radio networks: A contract-theoretic modeling approach," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 843–855, Apr. 2011.
- [10] Q. Liang, G. Sun, and X. Wang, "Random leader: A distributed-centralized spectrum sharing scheme in cognitive radios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 1806–1810.
- [11] D. Niyato and E. Hossain, "Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of Nash equilibrium, and collusion," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 192–202, Jan. 2008.
- [12] S. K. Behera, L. K. Behera, C. Madhusudana, and P. Padhi, "Spectrum scheduling for 3G and 4G networks by intra-operator flexibility," in *Proc. Int. Conf. Comput., Commun. Autom.*, May 2015, pp. 1391–1394.
- [13] P. Wang, B. Wang, W. Wang, Y. Zhang, and C. Wang, "Based multi-operator shared network opportunistic spectrum sharing," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst.*, vol. 2, Oct. 2012, pp. 864–868.
- [14] C.-H. Liu and H.-C. Tsai, "Traffic management for heterogeneous networks with opportunistic unlicensed spectrum sharing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5717–5731, Sep. 2017.
- [15] Z. Jiang and S. Mao, "Interoperator opportunistic spectrum sharing in LTE-unlicensed," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5217–5228, Jun. 2017.
- [16] S. Nakamoto. (2008). *Bitcoin: A Peer-To-Peer Electronic Cash System*. [Online]. Available: <http://bitcoin.org/bitcoin.pdf>
- [17] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
- [18] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "BlocHIE: A BLOCkchain-based platform for healthcare information exchange," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2018, pp. 49–56.
- [19] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 58241–58254, 2019.
- [20] Y. Wang, Z. Su, Q. Xu, and N. Zhang, "Contract based energy blockchain for secure electric vehicles charging in smart community," in *Proc. IEEE 16th Int. Conf. Depend. Autonomic Secure Comput. 16th Int. Conf. Pervasive Intell. Comput. 4th Int. Conf. Big Data Intell. Comput. Cyber. Sci. Technol. Congr.*, Aug. 2018, pp. 323–327.
- [21] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the Internet of Things," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 983–994, Jul. 2017.
- [22] S. Bayhan, A. Zubow, P. Gawlowicz, and A. Wolisz, "Smart contracts for spectrum sensing as a service," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 648–660, Sep. 2019.
- [23] S. Bayhan, A. Zubow, and A. Wolisz, "Spass: Spectrum sensing as a service via smart contracts," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Oct. 2018, pp. 1–10.
- [24] M. B. H. Weiss, K. Werbach, D. C. Sicker, and C. E. C. Bastidas, "On the application of blockchains to spectrum management," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 2, pp. 193–205, Jun. 2019.
- [25] K. Kotobi and S. G. Bilen, "Secure blockchains for dynamic spectrum access: A decentralized database in moving cognitive radio networks enhances security and user access," *IEEE Veh. Technol. Mag.*, vol. 13, no. 1, pp. 32–39, Mar. 2018.

- [26] E. Di Pascale, J. McMenamy, I. Macaluso, and L. Doyle, "Smart contract SLAs for dense small-cell-as-a-service," 2017, *arXiv:1703.04502*. [Online]. Available: <http://arxiv.org/abs/1703.04502>
- [27] G. Wood. Ethereum Project Yellow Paper. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, pp. 1–32. [Online]. Available: <http://www.cryptopapers.net/papers/ethereum-yellowpa-per.pdf>
- [28] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 254–269.
- [29] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 792–800.



**SHUANG ZHENG** (Student Member, IEEE) received the B.E. degree in electronic information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2018, where she is currently pursuing the master's degree with the School of Electronic Information and Communications. Her research interests are spectrum sharing and blockchain technology.



**TAO HAN** (Member, IEEE) received the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in December 2001. From August 2010 to August 2011, he was a Visiting Scholar with the University of Florida, Gainesville, FL, USA, as a Courtesy Associate Professor. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include wireless communications, multimedia communications, and computer networks. Dr. Han is serving as an Area Editor for the *EAI Endorsed Transactions on Cognitive Communications*.



**YUNA JIANG** (Student Member, IEEE) received the B.E. degree in communications engineering from the China University of Mining and Technology, Xuzhou, China, in 2017. She is currently pursuing the Ph.D. degree with the School of Electronic Information and Communications, Huazhong University of Science and Technology. Her research interests are the Internet of Things and blockchain technology.



**XIAOHU GE** (Senior Member, IEEE) received the Ph.D. degree in communication and information engineering from the Huazhong University of Science and Technology (HUST), China, in 2003. He has been worked with HUST, since November 2005. Prior to that, he worked as a Researcher at Ajou University, South Korea, and the Politecnico Di Torino, Italy, from January 2004 to October 2005. He is currently a full Professor with the School of Electronic Information and Communications, HUST. He is also an Adjunct Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Australia. He has published about 200 articles in refereed journals and conference proceedings. He has been granted about 25 patents in China. His research interests are in the area of mobile communications, traffic modeling in wireless networks, green communications, and interference modeling in wireless communications. He services as an IEEE Distinguished Lecturer and an Associate Editor for IEEE ACCESS, the IEEE WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. . . .