Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6639107, 10 pages
https://doi.org/10.1155/2021/6639107

Research Article

Blockchain-Based Trust Auction for Dynamic Virtual Machine Provisioning and Allocation in Clouds

Hao Xu,[1] **Weifeng Liu**,[1] **and Xu Liu**[1,2]

[1]*School of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225127, China*
[2]*School of Business, Victoria University, Melbourne VIC 3011, Australia*

Correspondence should be addressed to Xu Liu; sherryliu08@foxmail.com

Received 16 November 2020; Revised 21 April 2021; Accepted 18 May 2021; Published 16 June 2021

Academic Editor: Chunpeng Ge

Cloud computing uses virtualization technology to provide users with different types of resources in the form of services. The third party plays a crucial role in coordinating cloud market between cloud providers and users. As for providing services or trading, the extra broker fees are required for the middleman because the third party facilitates transactions. Moreover, there is no guarantee that the third party is trusted, which can lead to information leakage, data tampering, and unfair trading. Blockchain technology is an emerging technology that can store and communicate data between entities that unnecessarily trust each other. To resolve the problems, this paper presents the blockchain-based trust and fair system and develops the smart contract of auction and transaction. The prototype system is implemented based on the Hyperledger Fabric. The experimental results prove the feasibility of the scheme.

1. Introduction

Cloud computing is a popular paradigm of offering services over the Internet [1]. With the development of Internet technology and virtualization technology, more and more enterprises and individuals outsource their workloads to cloud providers. Cloud computing services are generally provided in three types: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [2]. In cloud computing environment, cloud providers cooperate to form a huge abstract, virtualized, and dynamically expandable resource pool to provide cloud service and resources to users. A user acquires and releases resources by requesting and returning virtual machines in the cloud. To sell the VM instances to users, cloud providers can employ auction-based models. An example of implemented auction method in Cloud computing is the spot market introduced by Amazon [3].

Current cloud resource auction integrates the network technology into the bidding system. The desirable preconditions for managing cloud resource auction are "trust" and "fair." The main entities included in cloud resource auction are cloud providers, users, and the third party (i.e., auctioneer) as shown in Figure 1. The cloud provider is an organization that offers computing resources for use on payment. A user is a person or an organization that purchases cloud service. The third party is a middleman that provides a platform to make the product available in the market. Most of the third-party platforms are centralized middlemen which can lead to a host of trust and fair issues. First, the centralized cloud computing auction system may suffer the single node attack and has a higher risk of data tampering and privacy leakage [4]. Secondly, the third-party platform completely controls over the bidding process. Users have no way to ensure that the middleman never leaks their bidding information. Even if the auction platform has security and trust issues, it still charges a large fee for service as a middleman between cloud providers and users. In addition, many existing researches and models on cloud resource auction cannot consider the transaction process after the auction. There are multiple unfair problems associated with the transaction process. For example, malicious cloud providers may not

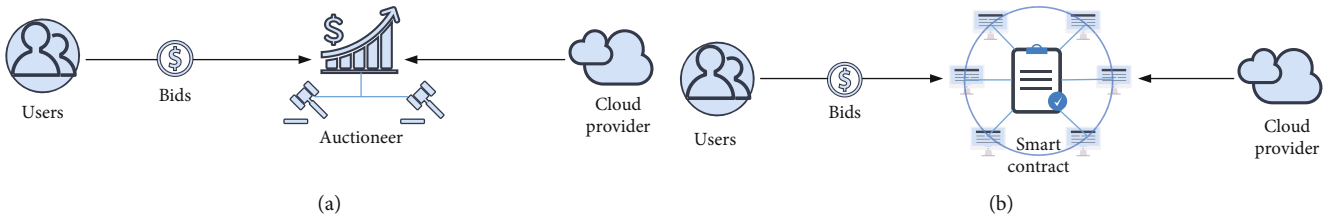

Figure 1: (a) Current centralized model of cloud resource auction. (b) Proposed architecture using smart contract which acts as a third-party platform.

provide services or unmatched resources after charging service fees, and malicious users may not pay service fees when obtaining resources.

Blockchain, as an emerging distributed ledger technology [5], has attracted more attention from various fields, including supply chain [6], IoT [7], medical [8], and other areas. Since data storage is based on a distribution architecture and data management is based on a peer-to-peer network, data records are immutable, verifiable, and traceable [9]. Not only blockchain provides a reliable system to store data but also it improves bidding information security and immutability. So blockchain can facilitate transactions between entities that unnecessarily trust each other.

Blockchain technology has the great potential to address the challenge of conventional cloud platform. References [10–12] mainly aim at trust problem in the cloud platform and design a credible service transaction method combined with blockchain. In [13], a decentralized framework is proposed to address issues such as trust and data security. A blockchain-based cloud resource allocation framework is proposed in [14]. Conspicuous feature, of this model, is the timed commitment scheme, state mechanism, and ladder payment to guarantee auction fairness and trade fairness.

In this paper, we combine blockchain and smart contract technology to propose a blockchain-based cloud resource auction scheme to ensure the security of bidding data and the fairness of cloud services. The scheme designs a data encryption transmission module, which uses symmetric encryption technology to encrypt bid data to protect user privacy and prevent malicious users from leaking data. Since the blockchain is a peer-to-peer access structure, the points in the structure can be trusted with each other. Consequently, the centralized third-party platform can be removed to reduce the transaction cost. Use smart contract technology to design the auction contract and transaction contract. Some rules are written in smart contracts to achieve fair trading of cloud resources. Users must pay transaction fees in advance to obtain VM instances, and cloud providers must provide cloud resources in order to receive transaction fees. Compared with the traditional cloud resource auction system, this scheme can realize the safe auction and fair transaction of cloud resources, protect the bidding privacy of users, and resist internal malicious users.

The structure of this paper is as follows. Section 2 introduces the background, system architecture and detailed implementations. Section 3 gives the results of the proposed architecture. The last section concludes the paper.

2. Background

2.1. Related Work. One helpful approach for the cloud computing resource supply is the mechanisms for auctions. Several authors have studied such mechanisms in different fields such as economics and computer science. In the context of cloud computing resource allocation, Jain et al. [15] proposed an efficient truthful-in-expectation mechanism for resource allocation in clouds. Zama and Grosu [16] designed truthful approximation mechanisms for the auction-based allocation of VM instances in clouds. Due to the characteristics of resources in cloud computing, more and more researchers introduce the quality of service parameter and nonprice attributes into cloud resource auction system to solve allocation problems and transaction fraud problems. In [17], the authors introduced a multiattribute auction framework and used evaluation functions to publish cloud providers for fraudulent behavior. The current works explore auctions of introducing features for the cloud resource auction system. However, none of the noticed solutions focus on the honest behavior of the auctioneer, nor in providing reliable records about transactions between the cloud provider and users.

In such a direction, to combine blockchain technology with the cloud resource auction system has become a new trend, because its decentralization and data tampering prevention might provide solutions to the shortcomings of current systems. Few works have been using blockchain to solve the cloud resource auction system issues. In one of them, Chen et al. [18] designed a blockchain-based architecture for a bidding system. Franco et al. [19] proposed a blockchain-based reserve auction for infrastructure supply in a virtual network. Also, An et al. [20] proposed a blockchain-based reserve auction for data transactions to solve the problem of third-party brokers. Ch et al. [21] presented a blockchain technology solution, using pentatope-based elliptic curve cryptography and SHA to improve the security and privacy device data. Thirumalai et al. [22] applied the knapsack method to encrypt ENPKESS keys to enrich high security in cloud systems.

2.2. Blockchain. Blockchain originated from Bitcoin. As the underlying application technology of Bitcoin, it has received widespread attention as Bitcoin became famous. As a decentralized distributed ledger technology, blockchain integrates technologies such as distributed data storage, encryption algorithms, and consensus mechanisms. Blockchain eliminates third-party authoritative centers [23]. It can enhance

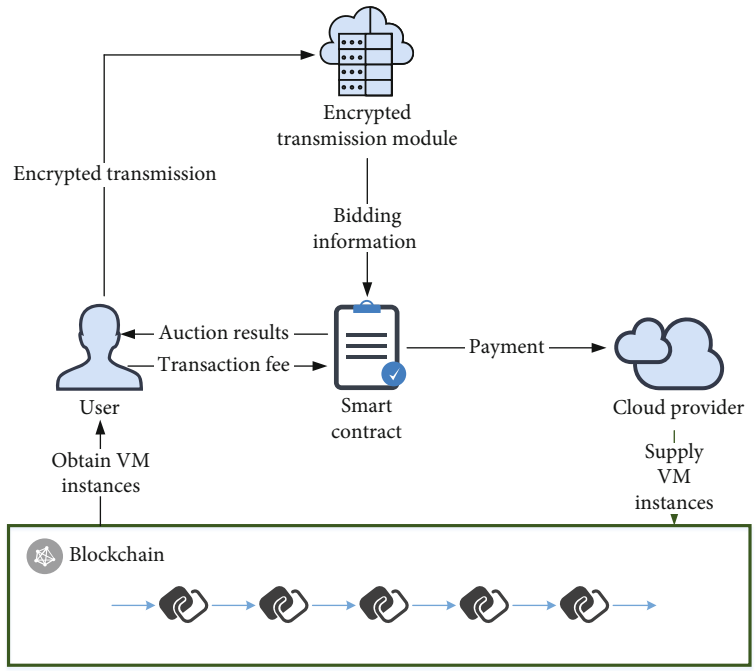

Figure 2: Architecture model.

the trust between participants without relying on the third-party trusted authority, because all consensus nodes participate in the process of maintaining blocks on the chain and the transaction record are stored on a peer-to-peer network to ensure that the data cannot be tampered with and any node can audit the data in the block. In addition, due to the irreversibility of the hash function and the security advantages of the consensus mechanism, the transparency, security, and immutability of transaction data are guaranteed.

Since the blockchain is public and transparent, when it comes to private data, it needs to be encrypted. This paper mainly uses *Diffie-Hellman key exchange algorithm* [24] and *symmetric encryption algorithm* [25]. The characteristic of the *DH algorithm* is that both parties can exchange keys securely in an insecure channel. In the *DH algorithm*, the sender and receiver use their own private key and the other party's public key to generate a shared key. In the *symmetric encryption algorithm*, the sender uses the symmetric encryption key to encrypt the data, and the receiver uses the key to decrypt the data to obtain the data, which is characterized by the same encryption and decryption keys.

In the blockchain 2.0 stage, the concept of smart contract [26] is introduced. Developers can create various applications through smart contracts. In this paper, the smart contracts replace the third-party platform in the existing system and perform resource allocation and transaction management according to predefined rules.

2.3. Proposed Model. This paper focuses on IaaS and is built under the situation that the cloud provider offers different types of resources in the form of VM instances. The blockchain-based cloud computing resource auction model is shown in Figure 2, including user, cloud provider, certificate authority, data encryption transmission module, and smart contract module.

The main function of the user is to obtain the VM instances. Each user has his own public and private key PK_{user}, SK_{user}, and address $Addr_{user}$. In the process of applying for obtaining the VM instances, the user can obtain the resources only after being authorized by smart contract. The main function of the cloud provider is to supply the VM instances. The cloud provider has his own public and private key PK_{cfp}, SK_{cfp}, and address $Addr_{cfp}$.

The certification authority is a role in the system, which is mainly responsible for verifying the identity of users who want to join the system and then sending digital certificates to users. Only users who pass identity verification can join the system, which improves the security of the system to a certain extent.

The data encryption transmission module encrypts each user's bid information and uploads it to the smart contract module to ensure that the bid information will not be leaked. The smart contract module realizes the security and fairness of the whole auction process and solves the problems such as auction centralization, bidding information disclosure, and dishonest transaction. The module is mainly composed of two smart contracts: auction contract and transaction contract. The auction contract (AC) executes allocation algorithm and payment function, which is aimed at social welfare maximization to protect the interests of users. Use the transaction contract (TC) to determine the applicant's transaction information before the transaction occurs, freeze the prestored transaction fee when the transaction occurs, and send the transaction fee to the cloud provider after the transaction is completed and the applicant receives the VM instances. Use smart contract to replace trusted third parties

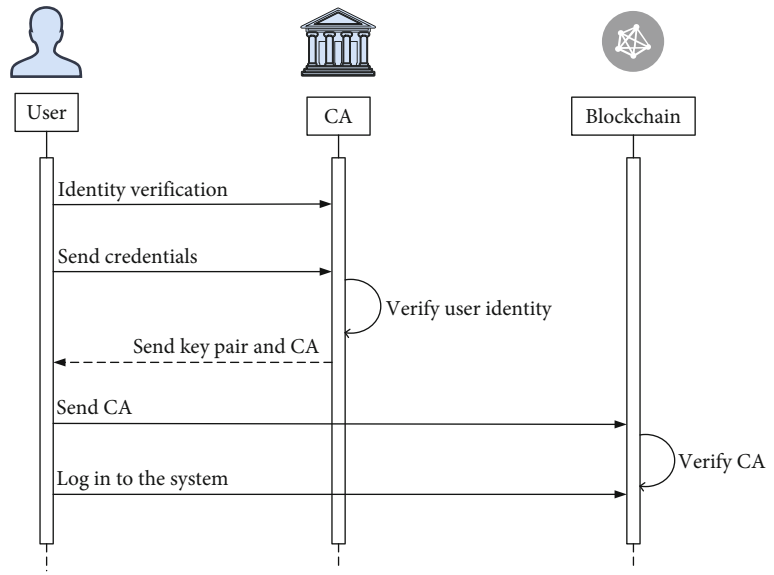

Figure 3: User registration process.

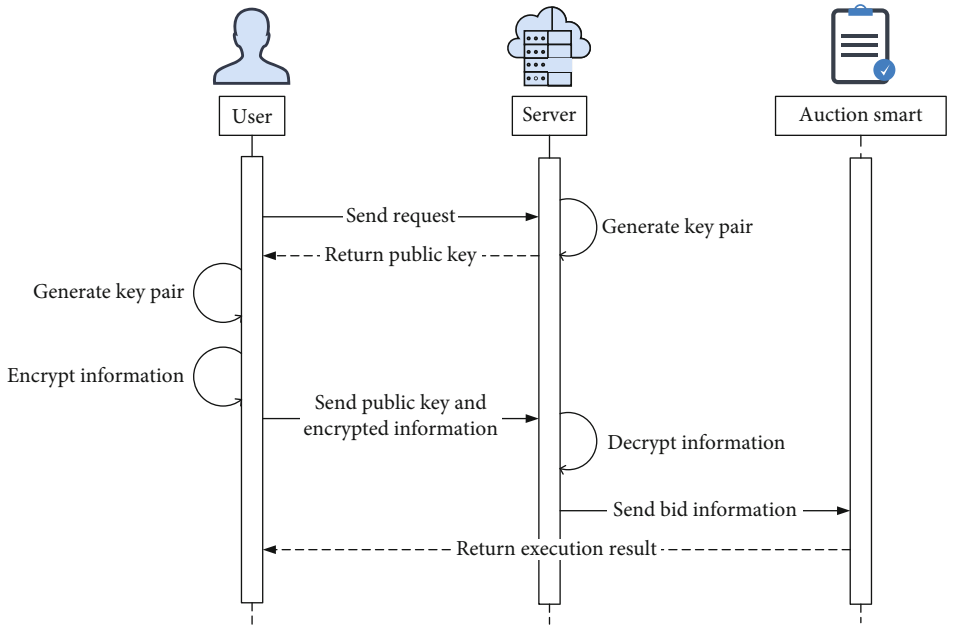

Figure 4: Timing diagram of encrypted transmission module.

to realize transaction payment management, effectively preventing users from not paying and sellers not providing resources, and ensuring fair transactions.

2.4. Blockchain-Based Cloud Computing Resource Auction Steps

2.4.1. User Registration. The process of identity authentication and registration is shown in Figure 3.

Step 1. The user needs to send an identity verification application to the certification authority (CA) before logging in to the alliance chain network. After the verification is passed, the applicant will obtain the key pair and a digital certificate.

Step 2. The user sends his registration and digital certificate to the system after obtaining the digital certificate and key pair.

Step 3. The node verifies the digital certificate of the new user and confirms the identity with the CA. After identity is approved, the user will be added to the system.

2.4.2. User Submission. The process of submitting bid data is shown in Figure 4.

Step 1. The user sends a request to the server to generate the public and private key on the server side. The server side returns the production key to the user.

Input: server public keyPK_{ser}, client public keyKEY_{cli}, encrypted bidding information$CInfo_{bid}$, actual submission timet_A, deadlineT_s, user A address$Addr_A$
Output: complete submission
1 KEY_{cli}= *client.getSecret* (PK_{ser})
2 $CInfo_{bid}$= *Enc* ($Info_{bid}$,KEY_{cli})
3 *Client node* (PK_{cli},$CInfo_{bid}$,t_A,$Addr_A$) → *Server node*
4 KEY_{ser}= *server.getSecert* (PK_{ser})
5 $PInfo_{bid}$= *Dec* ($CInfo_{bid}$,KEY_{ser})
6 ***if*** $t_A < T_s$ && $PInfo_{bid}$ = = *true* ***then***
7 *Server node* ($PInfo_{bid}$,$Addr_A$) → *AC*
8 ***return*** *"Bid information submission successful"*
9 ***else***
11 ***return*** *"Bid information submission failure"*
12 ***end if***

Algorithm 1: Encrypted submission.

Input: Users bidding information :$Info_{bid} = \{\theta_1, \theta_2, \cdots, \theta_n\}$, $\theta_n = (\beta_n, b_n)$
Users address information: $Addr_U = \{Addr_1, Addr_2, \cdots, Addr_n\}$.
Resources capacities: $\mathbf{C} = \{C_1, C_2, \cdots, C_M\}$
Output: $Addr^*$,x^*, P
1 {Collect}
2 for all $i \in U$ **do**
3 collect user bidding information and address from user i
4 {Allocation}
5 $(Addr_i, x_i) \leftarrow$ **G-VMPAC-II-ALLOC** (θ,**C**)
6 $Addr^* \leftarrow Addr^* \cup Addr_i$
7 $x^* \leftarrow x^* \cup x_i$
8 {Payment}
9 $P_i \leftarrow$ **G-VMPAC-II-PAY** (θ,**C**)
10 $P \leftarrow P \cup P_i$
11 return $Addr^*$, x^*, P

Algorithm 2: The function of AC.

Step 2. The user executes *DH algorithm* to generate public and private key pair and generate symmetric key according to the public key of the server, which is used to encrypt the transaction information.

Step 3. The user sends his public key and encrypted information to the server.

Step 4. The server generates a symmetric key according to the user's public key and decrypts the encrypted information by using the symmetric key.

Step 5. Send the bid information to the auction contract, and return the execution result to the user.

Algorithm 1 describes the process of the data encrypted submission. First of all, user A requests and obtains the public key of the server. Lines 1 to 2 indicate that user A generates the symmetric key KEY_{cli} according to the public key of the server and encrypts the bid information $Info_{bid}$ with KEY_{cli} to generate cipher text $CInfo_{bid}$. Line 3 shows that user A sends the public key of the client PK_{cli} and encrypted bid information $Info_{bid}$ to the blockchain server. Lines 4 to 5 show that the blockchain server generates the same symmetric key KEY_{ser} (KEY_{cli}=KEY_{ser}) and decrypts the $CInfo_{bid}$ using KEY_{ser} to obtain the details of the bid. Lines 6 to 11 show that the submission time limits and verifies whether the bid information has been tampered with. Finally, the server sends bid information to the AC.

2.4.3. Auction. This section presents the function of the auction contract. Mahyar et al. [27] proposed truthful greedy mechanism for VM provisioning and allocation in clouds. The mechanism is mainly divided into two parts: allocation and payment. The function of auction smart is described in Algorithm 2. Lines 1 to 3 show that AC collects the bidding information and address from the users. Lines 4 to 7 show the optimal allocation of solving cloud resource allocation by calling the allocation algorithm G-VMPAC-II-ALLOC [27]. Lines 8 to 10 show that the payment of user i is calculated by calling G-VMPAC-II-PAY [27]. Finally, AC returns three output parameters:$Addr^*$, the address set of winner;x^*, the optimal allocation of cloud resources to the users; and P the payment.

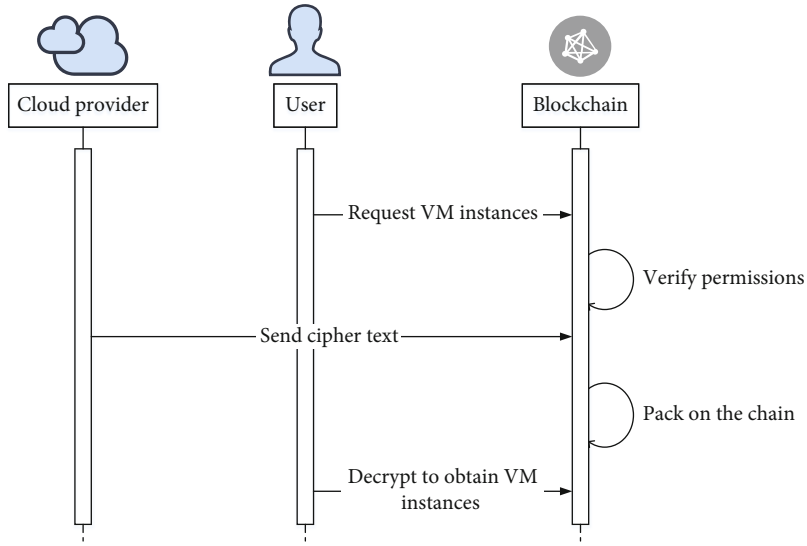

Figure 5: Resource supply process.

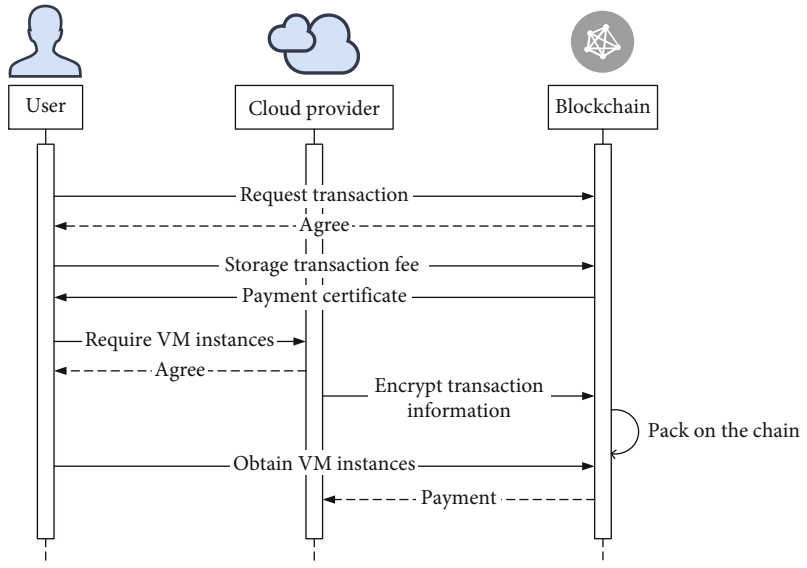

Figure 6: Pay service fee process.

2.4.4. Transaction Management. After the user receives the auction result, the winner can apply for resources. The TC needs to call the AC to obtain the address and the payment of the winner user to verify the applicant's transaction information. Transaction management is divided into two parts, including resource supply and payment management. The specific steps are shown in Figures 5 and 6.

(1) Resource Supply.

Step 1. Applicant A sends the request information; TC first confirms his payment certificate.

Step 2. If applicant A has payment certificate, the VM instance information will be encrypted with the applicant's public key and signs it with the cloud provider's private key. The cloud provider sends encrypted information and signatures to the network as transaction data.

Step 3. The node verifies the signature and packages the verified transaction data into blocks and waits for the consensus to be on the chain.

Step 4. After the block is on the chain, applicant A can obtain the VM instance information from the block. Applicant A verifies the signature and uses his private key to decrypt the transaction data to obtain the VM instance information.*(2) Payment Management.*

Step 1. Applicant A sends a request to TC for prestored payment. After TC receives the transaction fee, it will generate a

Input: Applicant A address$Addr_A$, applicant public and private keys(PK_A,SK_A), the CFP public and private keys(PK_{cfp}, SK_{cfp}), VM instances permission information$Info_{VM}$, blockID_{block}.
Output: Applicant A obtains virtual machine permission information
1 function*ResourceSupply* ($Addr_A$,PK_A,$Info_{VM}$,SK_{cfp})
2 $x_i \leftarrow$ *matchAC* ($Addr_A$, $Addr^*$)
3 if (x_i= = true) **then**
4 $CInfo_{VM}$= enc ($Info_{VM}$,PK_A)
5 return $CInfo_{VM}$
6 $\text{sign}_{cfp} \leftarrow \text{Sign}(hash(CInfo_{VM}), SK_{cfp})$
7 block $\leftarrow$ Add ($CInfo_{VM}$, sign_{cfp})
8 end function
9 function*GetInfo* (ID_{block}, SK_A, PK_{cfp})
10 sign_{cfp},$CInfo_{VM} \leftarrow$ *GetBlockInfo* (ID_{block})
11 validate (sign_{cfp},PK_{cfp})
12 $Info_{VM} \leftarrow$ dec ($CInfo_{VM}$,SK_A)

Algorithm 3: Resource supply.

Input: Applicant A address$Addr_A$, actual stored paymentss_A, actual stored timet_A, deadlineT_P, auction contract AC, CFP address $Addr_{cfp}$
Output: Complete transaction fee storage and transfer
1 User node (s_A, t_A,$Addr_A$) $\rightarrow$ TC
2 function TC(P, $Addr$, T_p)
3 $P_A \leftarrow$ *matchAC*($Addr_A$,P)
4 if $t_A < T_P$&&$s_A = P_A$ **then**
5 **return** o_{pay}
6 else if $t_A < T_P$&&$s_A < P_A$
7 **return** "*Insufficient transaction fee*"
8 else if $t_A > T_P$
9 **return** " *Storage failure*"
10 end function
11 function *Pay*(s_A,$Addr_{cfp}$)
12 $s_A \rightarrow Addr_{cfp}$
13 end function

Algorithm 4: Payment management.

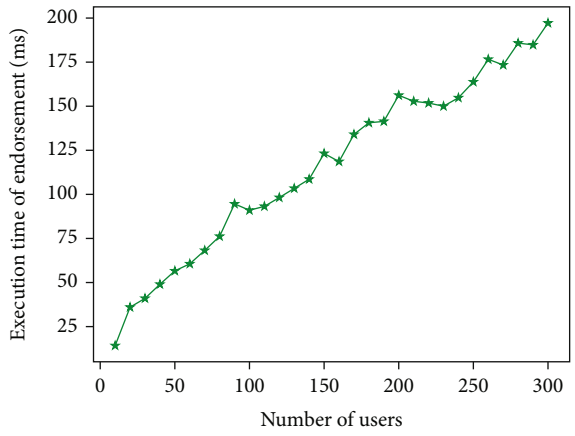

Figure 7: Execution time of auction in endorsing peer.

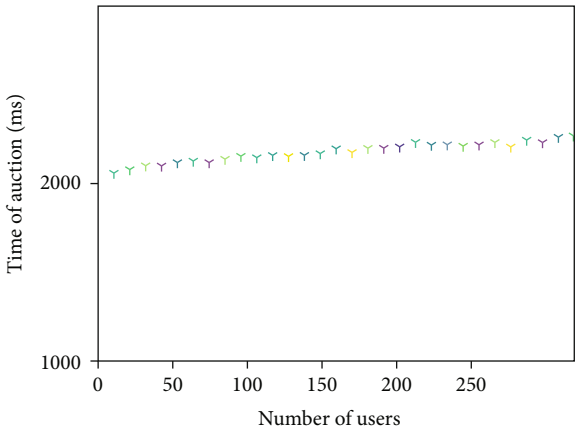

Figure 8: Time of auction phase in fabric.

Table 1: This work was compared with the models in the literature review.

Items	Models	Privacy protection	Decentralization for auction	Transaction management
[13]	Centralized	×	×	×
[14]	Centralized	×	×	√
[4]	Blockchain-based	×	√	×
This work	Blockchain-based	√	√	√

payment certificate o_{pay} and send it to applicant A, indicating that the prestored transaction fee has been completed.

Step 2. Applicant A sends the payment certificate, public key, and related information to the cloud provider.

Step 3. The cloud provider confirms the transaction and provide VM instances to applicant A.

Step 4. TC pays the service fee to the cloud provider after it receives the confirmation information that applicant A has obtained the virtual machine instance.

Algorithm 3 describes the process of the cloud provider providing resources to the applicant. Lines 2 to 5 show whether applicant A is the winner. If he is the winner, the cloud provider encrypts the resource information with his public key. Lines 6 to 9 show that the cloud provider publishes resource information on the blockchain. Lines 10 to 12 show that the applicant achieves VM instance permission information.

Algorithm 4 describes the payment management. Lines 2 to 10 show whether applicant A has stored enough money within a limited time. Lines 11 to 13 show that TC sends the transaction fee to the cloud provider.

3. Experiment

The simulation development of the system runs on a PC and uses the Ubuntu operating system. This solution is based on the open-source framework of Hyperledger Fabric v1.1 version. The system is simulated and developed through a cloud server. The development software uses Visual Studio Code and Remote Development plug-ins. The database Couchdb is used, and the system chain code is developed using node.js.

3.1. Performance Testing. The auction phase includes reading users' data sets, executing the auction allocation algorithm, and uploading the auction results to the chain. Endorsement node (endorser) simulates the execution of the allocation algorithm by calling the chain code; ordering service (orderer) sorts transaction requests and creates the block and peer updates ledgers after verification. In order to reflect the variation of auction time with the number of users, this performance test mainly studies the execution time of auction in endorsers and auction transaction time in fabric network. The system test results of the above two items are shown in Figures 7 and 8.

Figure 7 shows that the execution time of the allocation algorithm is affected by the increase in the number of users. As the number of users varies from 10 to 300, the execution time sees an upsurge from about 14 ms to 197 ms. By testing the auction execution time in the endorser, it is found that the increase in the number of users has a positive impact on the execution time. Figure 8 mainly tests the time of auction transactions in the fabric network. After many tests, each auction transaction time of the system is close to 2000 ms, which can meet the basic business needs.

Through the analysis of test results, the system has passed the functional test and performance test, completed the expected design goal, and verified the feasibility and effectiveness of the system.

3.2. Comparison. We compared the performance of our system with mainstream system. The result is presented in Table 1. In the table, the systems are compared on the three criteria:

(i) Privacy protection: in cloud auction, bidding prices and cloud resource demands are crucial for privacy. This system tackled privacy leakage through the data encryption module. This approach encrypts private information to ensure that no plain text appears during transmission, which improves data security

(ii) Decentralized for auction: it was already well known that decentralized systems, in which the smart contract acts as the third-party platform, were much more secure than centralized systems. Furthermore, to cut down the broker fees between users and cloud providers, the system should be based on the blockchain auction system

(iii) Transaction management: in the cloud market, both auction security and fair transactions are required. This proposed system eliminates malicious behavior in cloud resource transactions in which each user prestores transaction fees. This approach improves fairness, especially when the user refuses to pay or the cloud provider provides substandard cloud resources

4. Conclusions

This paper first introduces the difficulties and challenges of the current cloud computing resource auction system. Then, through the thorough analysis of the major issues of the system, we propose a cloud computing resource auction system based on blockchain for data security and transaction fairness. To ensure the privacy and integrity of bidding data, we design an encrypted transmission module, which uses

key exchange protocol and *symmetric encryption technology* to encrypt and transmit bidding data. In the transaction process, *asymmetric encryption technology* is used to encrypt the permission information of VM instance permission information to protect the rights and interests of users. Realize transaction fairness without a trusted third party through AC and TC. Transaction fairness realizes that users cannot receive VM instances without paying the corresponding service fees, and the CFP will certainly obtain the corresponding service fee after providing VM instances.

This paper is a preliminary exploration of applying blockchain technology to the cloud computing resource auction filed. There are still some shortcomings, including the following: (1) the honest bidding of users and the fair distribution of the system are mainly affected by the auction mechanism. Therefore, we should optimize the auction mechanism to improve the fairness of distribution and pricing; (2) this paper only discusses the situation where one cloud service provider provides only four resources. We can explore multiple cloud auction systems and study more versatile fairness solutions to adapt to multiple market environments.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, in part by the Key Research Projects in Education Informatization in Jiangsu Province under Grant 20180012, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX18_2366, in part by the Yangzhou Science and Technology Bureau under Grant YZ2018209 and Grant YZ2019133, in part by the Yangzhou University Jiangdu High-End Equipment Engineering Technology Research Institute Open Project under Grant YDJD201707, and in part by the Open Project in the State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, under Grant 1907.

References

[1] G. Vinu Prasad, A. S. Prasad, and S. Rao, "A combinatorial auction mechanism for multiple resource procurement in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 904–914, 2018.

[2] Z. Li M. Li et al., "A hierarchical cloud pricing system," in *2013 IEEE Ninth World Congress on Services*, Santa Clara, CA, USA, June-July 2013.

[3] "Amazon EC2 Instance Types," http://aws.amazon.com/ec2/instance-types/.

[4] 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," *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018.

[5] Y. Zhang, X. Xu, A. Liu, Q. Lu, L. Xu, and F. Tao, "Blockchain-based trust mechanism for IoT-based smart manufacturing system," *IEEE Transaction on Computational Social System*, vol. 6, no. 6, 2019.

[6] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, Kunming, China, June 2016.

[7] Z. Guan, Y. Zhang, L. Wu et al., "APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

[8] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, pp. 25–30, Vienna, Austria, August 2016.

[9] T. M. Fernandez-Carames P. Fraga-Lamas et al., "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[10] R. Li, T. Chen, P. Lou, J. Yan, and J. Hu, "Trust mechanism of cloud manufacturing service platform based on blockchain," in *Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, August 2019.

[11] Q. Wang, C. Liu, and B. Zhou, "Trusted transaction method of manufacturing services based on blockchain," *Computer Integrated Manufacturing Systems*, vol. 25, pp. 3247–3257, 2019.

[12] A. V. Barenji, Z. Li, and W. M. Wang, "Blockchain cloud manufacturing: shop floor and machine level," in *Smart SysTech 2018; European Conference on Smart Objects, Systems and Technologies*, Munich, Germany, June 2018.

[13] Y. Y. Xu and Y. Wang, "Research on blockchain in cloud manufacturing resource allocation," *Journal of Frontiers of Computer Science and Technology*, vol. 8, 2021.

[14] Z. Chen, W. Ding, Y. Xu, M. Tian, and H. Zhong, "Fair auctioning and trading framework for cloud virtual machines based on blockchain," *Computer Communications*, vol. 171, pp. 89–98, 2021.

[15] N. Jain, I. Menache, J. Naor, and J. Yaniv, "A truthful mechanism for value-based scheduling in cloud computing," *Throry of Computing Systems*, vol. 54, pp. 178–189, 2011.

[16] S. Zaman and D. Grosu, "A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds," *IEEE Transaction on Cloud Computing*, vol. 1, no. 2, pp. 129–141, 2013.

[17] G. Baranwal and D. P. Vidyarthi, "A truthful and fair multi-attribute combinatorial reverse auction for resource procurement in cloud computing," *IEEE Transaction on Service Computing*, vol. 12, no. 6, pp. 851–864, 2019.

[18] Y. Chen, S.-H. Chen, and I.-C. Lin, "Blockchain based smart contract for bidding system," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, Chiba, Japan, April 2018.

[19] M. F. Franco, L. Z. Granvile, L. Z. Granville, and B. Stiller, "BRAIN: blockchain-based reverse auction for infrastructure supply in virtual network functions-as-a-service," in *2019 IFIP*

Networking Conference (IFIP Networking), Warsaw, Poland, May 2019.

[20] B. An A. Liu et al., "Truthful crowdsensed data trading based on reverse auction and blockchain," in *Database Systems for Advanced Applications. DASFAA 2019. Lecture Notes in Computer Science, vol 11446*, G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds., Springer, Cham, 2019.

[21] R. Ch, G. Srivastava, T. R. Gadekallu, P. K. R. Maddikunta, and S. Bhattacharya, "Security and privacy of UAV data using blockchain technology," *Journal of Information Security and Applications*, vol. 55, article 102670, 2020.

[22] C. Thirumalai, S. Mohan, and G. Srivastava, "An efficient public key secure scheme for cloud and IoT security," *Computer Communications*, vol. 150, pp. 634–643, 2020.

[23] Q. Lin, H. Wang, X. Pei, and J. Wang, "Food safety traceability system based on blockchain and EPCIS," *IEEE Access*, vol. 7, pp. 20698–20707, 2019.

[24] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transaction on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[25] G. J. Simmons, "Symmetric and asymmetric encryption," *ACM Computing Surveys*, vol. 11, no. 4, pp. 305–330, 1979.

[26] S. Omohundro, "Cryptocurrencies, smart contracts, and artificial intelligence," *AI Matters*, vol. 1, no. 2, pp. 19–21, 2014.

[27] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE Transaction on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 594–603, 2015.