Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1137130, 10 pages
https://doi.org/10.1155/2022/1137130

Research Article

Housing Rental Scheme Based on Redactable Blockchain

Chunli Wang,[1,2] **Wensheng Jia**,[1] **and Yuling Chen**[1,3]

[1]*State Key Laboratory of Public Big Data, College of Mathematics and Statistics, Guizhou University Guizhou Guiyang, 550025, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, 541000, China*
[3]*Blockchain Laboratory of Agricultural Vegetables, Weifang University of Science and Technology, Weifang, Shandong 262700, China*

Correspondence should be addressed to Wensheng Jia; wsjia@gzu.edu.cn

Received 19 December 2021; Revised 25 January 2022; Accepted 8 February 2022; Published 10 March 2022

Academic Editor: Jinguang Han

With the increase of the housing population, it is very important for renters to find credible housing information through rental platforms. However, due to the imperfection of relevant laws and insufficient supervision of the housing rental market, problems have emerged one after another. In this article, we propose a housing rental scheme based on redactable blockchain, introducing a trusted third party to verify the homeowner rental listings, and fundamentally preventing false listings from flowing into the housing rental market. Specifically, a trusted third party similar to the housing authority will issue a verifiable claim for the homeowners listing, and then the accounting node of the consortium blockchain will publish the verified listing information on the blockchain so as tenants to make inquiries according to their needs. At the same time, we use the new chameleon hash to build a redactable blockchain. After the homeowner proposes to change the housing price or housing information, the content of the historical block is modified as required. Thus, to a certain extent, the changes in housing prices can be monitored to curb chaos in the housing rental market.

1. Introduction

With the development of social economy and changes of people's concepts, not only the housing rental market's demand has increased significantly but also the number of renters has increased year by year. Nevertheless, the housing rental market is experiencing frequent occurrences. For example, the proliferation of false listings, the asymmetry of information in the leasing market, the serious leakage of customer information, and the low efficiency of industry transactions have caused chaos in the housing leasing market. The abovementioned problems have severely affected the vigorous development of the housing rental market and increased the additional burden of renters. Solving the various problems in the housing rental market plays a very important role in meeting the housing needs of residents and promoting stable economic and social development.

Therefore, we propose a housing leasing program based on blockchain. The blockchain is essentially a decentralized database. There is no centralized third-party organization in the blockchain system. Blockchain transactions are generated by participating entities. The transactions are packaged into blocks and added to the blockchain by miners in chronological order [1]. Participating entities in the blockchain need to update the block regularly and store it. Public chains are mature in Bitcoin and Ether, but private chains have emerged due to the real demand. There are selfish mining attacks in blockchain where attackers compromise the blockchain system by exploiting the vulnerability of the consensus mechanism [2]; while semiselfish mining attacks are proved by Li et al. through simulation that semiselfish mining is impossible in practice [3]. At present, blockchain has been used in many scenarios such as medical care, finance, and the Internet of Things. However, the use of blockchain in these scenarios cannot be directly applied to the housing rental market.

In the blockchain, information is disseminated according to the public key and private key, an asymmetric digital

encryption technology to achieve mutual trust between the two sides of the transaction. The public key and the key appear in pairs, and the public key is broadcasted for other users to know, while the private key is kept by the users themselves. In terms of public key setting, Chen et al. proposed a dynamic multikey FHE scheme based on the LWE assumption of public key setting [4], which uses the product of the public key and the product of a uniform random matrix to hide the secret key and improve the security of the public and private keys.

Accordingly, based upon the foregoing, we propose a plan to combine the blockchain with the housing leasing market to solve various problems in the leasing market. By introducing a trusted third-party verification agency, a weakly centralized blockchain is built to solve the problems of information asymmetry, illegal subletting, and tenant information leakage in the housing rental market. The homeowner submits the information of the house to be rented to a trusted third party for verification. The third-party agency verifies the housing information to ensure the authenticity and reliability of the rental housing and sends a verifiable certification claim to the verified homeowner. The homeowner sends the basic information and certification claim of the rented house to the accounting node. The accounting node verifies the certification claim and publishes the verified housing rental information on the blockchain for tenants to query. The tenant can query the rental information in the blockchain by paying a certain token RentCoin. When the homeowner and the tenant reach an agreement on renting a house, they automatically start the transaction by triggering a smart contract. After the lease transaction is generated, the homeowner and tenant pay a certain amount of RentCoin to the consortium blockchain nodes as a transaction fee. In this process, the tenant's information is only known to the homeowner. So, there is no leakage of the tenant's information.

The main contributions of this paper can be summarized as follows:

(1) We propose a scheme for house leasing based on blockchain technology. By introducing a trusted third party, we ensure the authenticity of rental information. Solve the problem of information asymmetry between homeowners and tenants in the housing rental market. In addition, our scheme has no housing intermediary, which saves the cost of housing intermediary

(2) The introduction of smart contract technology has improved the fairness of house leasing transactions. The conditions for triggering the smart contract are met, then the transaction is automatically executed, which improves the fairness of industry transactions

(3) Introduce a new chameleon hash to the consortium blockchain to reduce user interaction and realize the editability of the consortium blockchain

The rest of this paper is organized as follows. We review related work in Section 2. Section 3 introduces the relevant background knowledge that will be used in this article. We introduce the system model, threat model, and design goals in Section 4. Section 5 describes our proposed scheme in detail. Finally, Section 6 concludes the article.

2. Related Work

The chameleon hash and signature were proposed by Krawczyk and Rabin in 2000 to prohibit the recipient from freely disclosing the content of the signature information to any third party without the consent of the signatory [5]. But in the scheme proposed by Krawczyk and Rabin, there is the problem of key exposure. Therefore, some people have proposed whether it is possible to construct a chameleon hash and signature without key exposure. In 2004, Chen et al. first proposed an identity-based keyless exposure scheme. In this scheme, the author introduces a three-trapdoor mechanism, each transaction has its own trapdoor key, and no third party can create a trapdoor key that has not appeared before. Therefore, this scheme does not have the problem of key exposure [6].

In 2008, Nakamoto explained the blockchain technology in the article Bitcoin: A Peer-to-Peer Electronic Cash System [7]. Since blockchain technology was proposed, it has been applied to many fields such as vehicle transportation, securities finance, medical, and health care. The application of blockchain technology in specific scenarios is also becoming more mature. For example, Kang and others applied blockchain to vehicle edge computing and secure data sharing in 2019. Using consortium blockchain and smart contracts, the author built a secure vehicle data storage and sharing system [8]. Subramaniam et al. used blockchain technology to prevent credit fraud in 2020 and combined with Near Field Communication (NFC) to demonstrate the effect of this scheme [9]. Xu and others proposed a blockchain-based large-scale health data privacy protection scheme in 2019. The solution introduces the star file system (IPFS) to solve the storage problem of large-scale health data and realizes the privacy protection of health data based on blockchain technology. The author fully explained the program and gave the specific details of its implementation [10]. Chen et al. proposed a source location privacy (SLP) protection scheme (PSSPR) based on sector domain phantom routing in WSNs in 2021. The scheme has good performance in security [11].

In 2019, Ashritha et al. applied chameleon hash to blockchain and proposed redactable blockchain for the first time [12]. The introduction of chameleon hash can modify the content of the block without changing the block hash and other block contents. The master key in this scheme is dispersed among the master nodes based on secret sharing. When the content of a block is to be modified, the master node holding the master key share reconstructs the master key by secure multiparty computation to achieve the modification of the block content. In 2020, Xu et al. use editable blockchain for the management and authentication of mobile network identity. The article enables users to securely master their personal identity information by introducing self-sovereign identity. What is more, it empowers easier

and faster identity verification between users and network operators through editable blockchain and permits operators to dynamically revoke users through chameleon hashing [13]. The proposed redactable blockchain provides more scenarios for the practical application of blockchain.

In this paper, we introduce a new chameleon hash to achieve the editable character of the consortium blockchain as a way to build a housing rental platform. The method can solve the problems of information asymmetry and privacy leakage brought by the housing rental market to a certain extent.

3. Preliminaries

In this section, we introduce the background knowledge that will be used in this paper, including secret sharing, identity-based encryption, the new chameleon hash algorithm, and smart contracts.

3.1. Secret Sharing. Secret sharing was first proposed by Sharmir and Blakley in 1979 as the rational distribution of shared secrets among a group of users in order to achieve a shared ownership of the secret by all group members [14, 15]. Later in 1985, Chor et al. introduced the concept of verifiable secret sharing [16]. The concept was proposed considering the existence of dishonest participants in the secret sharing scheme, and it added some public commitment and verification algorithms to the secret sharing scheme as a way to detect dishonest users falsifying their secret shares [17–19]. Verifiable secret sharing assumes the existence of a secret S that is divided in a specific way into n shares $S_1, S_2, \cdots, S_n$ and securely given to n individuals for safekeeping, and the algorithmic process is the following two steps.

(1) *Secret Distribution Algorithm* $Share(S) = (S_1, S_2, \cdots, S_n)$. Given a secret S, n shares are randomly generated in a specific way

(2) *Secret Recovery Algorithm* $Rec(S_1, S_2, \cdots, S_n) = S \cup \perp$. Given the secret share $S_1, S_2, \cdots, S_n$, recover the secret S or return the outlier $\perp$

A (k, n)-verifiable secret sharing scheme needs to satisfy the following two requirements.

(1) *Verifiability*. A user can test whether a secret share is a valid share after receiving it. If the share is valid, the secret recovery algorithm can output a unique secret S. Any k valid shares out of n shares or more than k valid shares can recover the secret S

(2) *Unpredictability*. For polynomial-time algorithms, any less than k valid shares among n shares cannot fully recover the secret S

If k is larger, the higher the security of the secret sharing scheme. When $k = n$, all the secret sharers need to reconstruct the secret S together.

3.2. Self-Sovereign Identity. Self-sovereign identity (SSI) was proposed by Toth and Anderson-Priddy in 2019 [20], where each identity is fully owned, controlled, and managed by the owner of the identity. By virtue of a self-sovereign identity, users have full control over how their personal information is kept and used. Users with SSIs can store their data locally without relying on a central data repository. A service provider or organization can only access information about a user with the consent of SSI's owner. Thus, self-sovereign identity provides users with added security and flexibility.

3.3. Chameleon Hash. The chameleon hash was proposed by Krawczyk and Rabin in 2000 to prohibit the recipient from disclosing the contents of a signed message to any third party at will without the consent of the signer. The chameleon hash function is a cryptographic hash function that contains trapdoor information. The manager who has the trapdoor information can generate hash collisions based on the trapdoor information [1]. The chameleon hash satisfies the following security requirements.

(1) *Collision Resistance*. When inputting the public key hk, there is no effective algorithm to find the pair (m_1, r_1), (m_2, r_2) such that $\text{Hash}(hk, m_1, r_1) = \text{Hash}(hk, m_2, r_2)$, where $m_1 \neq m_2$

(2) *Trapdoor Collision*. At the input of trapdoor key tk, there exists a valid algorithm for any m_1, r_1, given m_2 can find r_2 such that $\text{Hash}(hk, m_1, r_1) = \text{Hash}(hk, m_2, r_2)$

(3) *Semantic Security*. For arbitrary messages m_1、m_2, the probability distributions of $\text{Hash}(hk, m_1, r_1)$ and $\text{Hash}(hk, m_2, r_2)$ are indistinguishable. In particular, for a randomly chosen r, no information about m can be inferred from $\text{Hash}(hk, m, r)$

Ateniese et al. replaced the hash of the block header in the blockchain with a chameleon hash to make the blockchain editable [21]. In 2018, Li et al. proposed a new chameleon hash for consortium blockchains, which gives each user of the consortium blockchain the right to modify the historical blocks without the need for multiparty interaction. The modification can be completed when the conditions for modification triggering are satisfied, i.e., a user is randomly selected according to the rules [22]. The random number $r = (r_1, r_2, \cdots, r_n)$ in the new chameleon hash function and the trapdoor key $(x_1, x_2, \cdots, x_n)$ are held by n users $P_1, P_2, \cdots, P_n$ in the consortium chain, respectively. The public keys of the n users are $(HK_1, HK_2, \cdots, HK_n)$, respectively. The new chameleon hash function is constructed as follows.

(1) $Setup(\lambda)$. Input security parameter λ, construct large prime p, q satisfying security parameter λ, where $p = kq + 1$, select element g of order q in multiplicative cyclic group Z_p^*, output public parameter $pp = (p, q, g)$

(2) $KeyGen(pp)$. Input the public parameter pp, randomly select the indices $x_1 \in Z_q^*, x_2 \in Z_q^*, \cdots, x_n \in Z_q^*$

, and calculate $h_1 = g^{x_1}, h_2 = g^{x_2}, \cdots, h_n = g^{x_n}$. Then the trapdoor private key $TK = (x_1, x_2, \cdots, x_n)$, the public key $HK = (g, h_1, h_2, \cdots, h_n)$

(3) $Hash(HK, m, r)$. Input hash public key $HK = (g, h_1, h_2, \cdots, h_n)$, message m and random number $r = (r_1, r_2, \cdots, r_n)$, output chameleon hash $CH = g^m h_1^{r_1} h_2^{r_2} \cdots h_n^{r_n} modp$

(4) $Forge(TK_i, m, r, m')$. Input trapdoor private key $TK_i = x_i$, message m, random number $r = (r_1, r_2, \cdots, r_n)$, message m'. Then according to $CH = g^{m'} h_1^{r_1} \cdots h_i^{r_i'} \cdots h_n^{r_n} modp$, we can get $m + x_1 r_1 + \cdots + x_i r_i + \cdots + x_n r_n = m' + x_1 r_1 + \cdots + x_i r_i' + \cdots + x_n r_n modq$, and we can calculate that $r_i' = (m - m' + x_i r_i) x_i^{-1} modq$

A user P_i with trapdoor key x_i can run the forge algorithm to find collisions such that $\text{Hash}(HK, m, r) = \text{Hash}(HK, m', r')$. Applying the new chameleon hash to the consortium blockchain can effectively reduce the interaction between users. Because each user in the consortium chain has the trapdoor key of the chameleon hash function, each can use their own trapdoor key to compute the new random number r' corresponding to the message m such that $\text{Hash}(m, r) = \text{Hash}(m', r')$. However, the modified block published by the user must satisfy the following two conditions to be accepted: (1) more than half of the users in the system agree to the modification, i.e., it contains the votes and signatures of more than half of the users; (2) the user who modifies the block is indeed the one corresponding to the smallest hash value obtained by computing the Lagrange interpolation formula.

3.4. n Noncooperative Game. Let $N = 1, \cdots, n$ be the set of insiders, $\forall i \in N$. The pure strategy set of insider i is the finite set $S_i = s_{i1}, \cdots, s_{im_i}$, and the mixed strategy set is $X_i = \{x_i = (x_{i1}, \cdots, x_{im_i}) : x_{ik_i} \geq 0, k_i = 1, \cdots, m_i, \sum_{k_i=1}^{m_i} x_i k_i = 1\}$. When each inning i chooses the pure strategy $s_{ik} \in S_i$, $i = 1, \cdots, n$, the inning i gets paid as the real number $R_i(s_1 k_1, \cdots, s_n k_n)$, and note that $X = \prod_{i=1}^{n} X_i, \forall x = (x_1, \cdots, x_n) \in X$. When each inning i chooses a mixed strategy $x_i = (x_{i1}, \cdots, x_{im_i}) \in X_i$ (i.e., inning i chooses the pure strategy $s_{i1}, \cdots$, with probability x_{i1}, and the pure strategy s_{im_i} with probability x_{im_i}) with $i = 1, \cdots, n$, and assumes that their choices are independent, then inning i gets an expected payoff that is real

$$f_i(x_1, \cdots, x_n) = \sum_{k_1=1}^{m_1} \cdots \sum_{k_n=1}^{m_n} R_i(s_1 k_1, \cdots, s_n k_n) \prod_{i=1}^{n} x_i k_i. \quad (1)$$

$\forall i \in N$, noting $\hat{i} = N \setminus \{i\}$. Each inning is rational and wants to get the maximum benefit for itself. Therefore, if there exists $x^* = (x_1^*, \cdots, x_n^*) \in X$ such that $\forall i \in N$, there is $f_i(x_i^*, x_{\hat{i}}^*) = \max_{u_i \in X_i} f_i(u_i, x_{\hat{i}}^*)$. Then x^* the Nash equilibrium point of this n-person noncooperative game, at which point each inning cannot make itself more profitable by individually changing its strategy [23, 24].

3.5. Smart Contract. Smart contracts are an idea first proposed by Szabo in 1994 and published on the website of the Extropy Institute [25]. A smart contract is defined as an event-driven, stateful program that runs on top of a replicated and shared ledger that holds the assets on the ledger [26]. Smart contracts can improve the fairness of housing lease transactions by simplifying issues such as the signing of housing lease contracts and subsequent defaults.

4. System Model, Threat Model, and Design Goals

4.1. System Model. As shown in Figure 1 below, the specific description is shown as follows.

4.1.1. House Owner. The homeowner is the owner of the home. They certify the home for rent through a certification agency. A landlord may have more than one home to rent, so a landlord can have multiple verifiable certification statements at the same time. The innkeeper stores the relevant certification statements locally and presents them to the tenant as the tenant needs them.

4.1.2. Tenant. Tenants query the blockchain to get the desired property information. And sign a lease contract with the landlord based on smart contract.

4.1.3. Certification Body. Certification bodies are distributed trusted entities, such as housing authorities. They issue a verifiable certification statement against the homeowner's listing that includes the certification authority's signature for others to verify. There is a relationship of trust between the certification authority and the homeowner. All the certification bodies form a consortium to maintain the consortium blockchain. When the owner pays the appropriate number of tokens RentCoin, the certification body can modify the contents of the blockchain with trapdoor information according to the owner's needs while keeping the block hash value unchanged.

4.1.4. Blockchain. It is a consortium blockchain maintained by accounting nodes for publishing the listing information and verifiable authentication claims of the homeowner. Any tenant can read the information on the blockchain. The transactions in each block form the Merkle hash tree of the respective block, where the first level of the hash tree is a new chameleon hash. An accounting node can modify the content of a transaction while keeping the block header unchanged.

4.1.5. Accounting Node. This node is a special node in the consortium blockchain where a trusted third-party acts as the accounting node for the consortium chain. It verifies the validity of the signatures of verifiable claims issued by the certification authority. At each time period, the consortium blockchain consisting of certification authorities chooses a leader, which is rotated by the certification authorities. The leader needs to pack valid transactions from the homeowner, generate new blocks, and join them to the consortium blockchain.

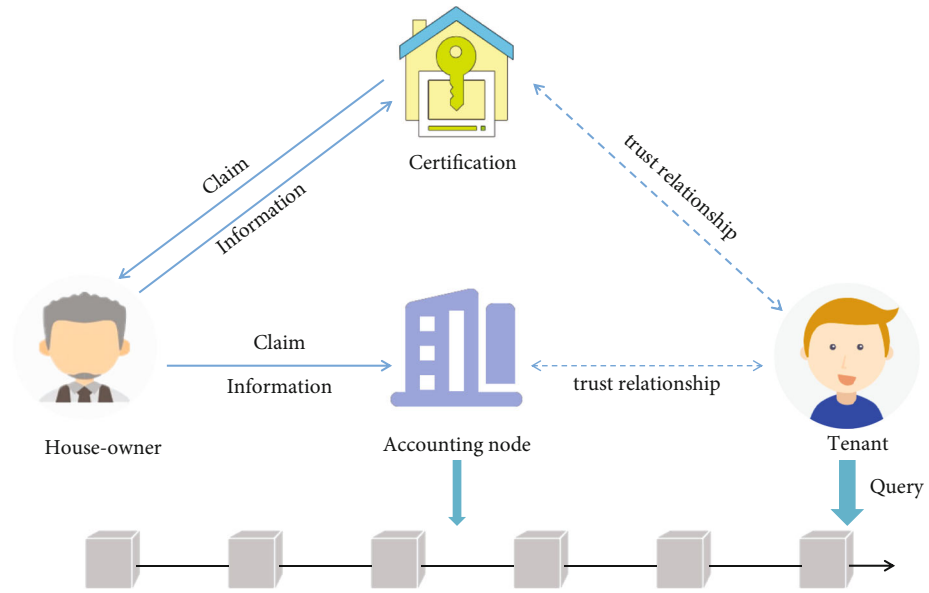

Figure 1: System model.

First, the homeowner sends the encrypted house information to the relevant certification authority according to the demand for rental. Second, the authentication agency decrypts and authenticates the information of the house. It also sends a verifiable claim to the authenticated owner. Finally, the owner sends the encrypted verifiable claims and the rental information of the house to the accounting node of the consortium blockchain. The leader of the consortium chain broadcasts the legitimate transaction to the other accounting nodes after verification. After they verify the signature, the leader adds the rental information to the consortium blockchain as a transaction for tenants to query and search.

4.2. Threat Model. We assume that a secure channel exists between the individual nodes. The certification authority strictly enforces the certification statement and authenticates truthfully. There are active and passive adversaries in the system. Passive adversaries obtain transaction data by eavesdropping on homeowners or analyzing leases between homeowners and tenants. Unlike passive adversaries, active adversaries may interrupt lease transactions between homeowners and tenants.

In addition, we assume that there are $3f + 1$ accounting nodes in the consortium chain to maintain the blockchain and that there are no more than f malicious nodes in the consortium chain.

4.3. Design Goals. Our goal is to achieve privacy and security, information symmetry, transaction fairness, and legal subletting in the rental market. Therefore, our design objectives are shown below.

4.3.1. Confidentiality of Property Information. The owner sends the ownership and other information of the house to the verification agency through a secure channel, and no third party can get the encrypted house information in the process, i.e., no agent can get the information of the rental property.

4.3.2. Authorized Home Ownership Certification. The certification claim issued by the certification agency must determine the match and authenticity of the homeowner's identity information and the housing information. It ensures the authenticity of the housing information as well as reduces the information asymmetry between homeowners and tenants.

4.3.3. Transaction Security. The rental transactions between landlords and tenants are only accessible to the house owner and dwellers themselves. No counterparty has access to the transaction information between them. There is also no third party to modify the transaction without the consent of the owner and tenant.

4.3.4. Privacy. Only the landlord and renter know their private information during the transaction. No third party has access to their private information without their permission.

4.3.5. Accountability. The rental contract signed by the leaseholder and landlord is based on a smart contract. They must be responsible for the contract they have signed and cannot break or deny the contents of the contract.

4.3.6. Legality of Subletting. If the landlord agrees to the tenant's subletting of an unexpired property, the house owner uses his or her private key to grant the boarder the legal right to sublet.

5. Proposed Scheme: Rentchain

5.1. Details of Our Proposed Scheme. Homeowners generate their own self-sovereign identity ID and corresponding public-private key pairs (pk, sk) according to their needs for rental properties, and get a legal verifiable certificate

statement through a trusted third-party organization. The owner sends the verifiable statement and *ID* to the accounting node of the consortium chain for verification. The accounting node publishes the information of rental properties with verifiable statements on the consortium chain for tenants to inquire according to their rental needs. If users in the consortium chain initiate a change request, they need to modify the content in the history block to initiate the voting phase. When more than half of the signatures of users in the consortium chain agree to the modification, the signatures of users who agree to the modification are broadcasted.

The user who changes the block is selected according to the distributed random generation (DRG) protocol and the Lagrange interpolation formula. The selected user modifies the content of the history block according to his own trapdoor key, and then broadcasts it to other users after the modification is completed. Only after all other users pass the verification, the changed history blocks are recorded and marked. The user who changes the block is selected according to the distributed random generation (DRG) protocol and the Lagrange interpolation formula. Selected user modifies the content of the history block according to his own trapdoor key and then broadcasts it to other users after the modification is completed. Only after all other users pass the verification, the changed history blocks are recorded and marked.

5.2. Rental Housing Information Up-Link Stage

(1) The homeowner H_i uses the *RSA* public key algorithm to generate his own identity ID_{H_i} and the corresponding public key pk_{H_i} and private key sk_{H_i}. Landlords H_i can generate different self-sovereign identity *IDs* and corresponding public-private key pairs according to their needs for renting out their properties. Tenant T_i uses *RSA* public key algorithm to generate their own public-private key pair (pk_{T_i}, sk_{T_i})

(2) The accounting node of the consortium chain (a trusted third-party verifier C_i) generates the corresponding public key hk_{C_i} and trapdoor private key tk_{C_i} based on the new chameleon hash Algorithm 1

(1) The landlord sends the basic information of the title deed, the self-sovereign identity *ID*, and the public key $(ID_{H_i}, pk_{H_i}, \inf_{H_i})$ of the house to be rented to the certification authority for authentication through a secure channel after encryption. The certification authority verifies the information after decrypting it using its own private key. If the information is incorrect, the certification authority rejects the request. Otherwise, the certification authority generates the verifiable statement claim and $\sigma_{H_i} = Sig_{sk_{C_i}}\{H(ID_{H_i}, pk_{H_i}, \inf_{H_i}), \text{claim}, t_1\}$, where t_1 is the verification period of the verifiable statement. Then, the certification authority C_i sends the verifiable statement and signature $\{ID_{H_i}, pk_{H_i}, \inf_{H_i}, claim, \sigma_{H_i}\}$ to the homeowner H_i through a secure channel. The homeowner needs to pay a certain amount of RentCoin to the certification authority as the certification fee

(2) After receiving a signed and authenticated statement from a certification authority, the owner requests the leader in the consortium chain to add the rental information to the chain. The owner encrypts a verifiable statement $E\text{claim} = Enc(ID_{C_i}, \{\text{claim}, t_1\})$ using the public key of the leader ID_{C_i} and sends it. After receiving the request from the homeowner, the leader first verifies whether the timestamp t_1 is within the allowed range compared to the current time. If not, the leader rejects the request; or else, the leader decrypts it with its own private key to get the verifiable statement claim

(3) The leader verifies the signature. If the signature is invalid, he rejects the request; contrarily the leader broadcasts the transactions for that period to other accounting nodes. After other accounting nodes verify the signature, the leader packages the signed transactions into blocks to join the consortium blockchain

Blockchain is a distributed system, which does not build upon a central authority. In our scheme, the consensus of the chosen consensus chain is Practical Byzantine Fault Tolerance (PBFT) [27]. We presuppose that there have $3f + 1$ accounting nodes in the consortium chain. There is only one leader for a time period, and the leader is rotated by the accounting nodes.

At regular intervals, the leader verifies the validity of the signature of the certification statement submitted by the homeowner. Before adding the transactions to the consortium blockchain, the leader broadcasts the results of the validation of the transactions. Only after getting the signatures of other accounting nodes, the leader packages the transactions for that period of time and adds them to the consensus blockchain. In this scenario, the hash of the transactions we use is the new chameleon hash. The new chameleon hash is used in order for the accounting node holding the trapdoor key to change the transaction content of the block on demand, while leaving the hash of the block unchanged. As seen in Figure 2, the first level of the Merkle tree is the new chameleon hash *h*, which is included in the transaction.

5.3. Smart Contract-Based Housing Rental Phase

(1) Tenants inquire and search for property information in the consensus chain according to their rental needs. If the landlord and tenant reach an agreement on the rental, the rental transaction is to be written into the consensus blockchain. The transaction includes rent, rental time, deposit, penalty for breach of contract, smart contract trigger conditions, and so on

Input: Security parameter κ;
Output: Secret trapdoor key$TK = (x_1, x_2, \cdots, x_n)$and public hash key $HK = (g, h_1, h_2, \cdots, h_n)$;
1. Select prime p, q, where $p = kq + 1$
2. Select the element g of order q in the multiplicative cyclic group Z_p^*
3. Select a random value $x_1 \in Z_q^*, x_2 \in Z_q^*, \cdots, x_n \in Z_q^*$ as the secret trapdoor key $TK = (x_1, x_2, \cdots, x_n)$
4. Compute$h_1 = g^{x_1}, h_2 = g^{x_2}, \cdots, h_n = g^{x_n}$, and set public hash key $HK = (g, h_1, h_2, \cdots, h_n)$;
5. return $TK = (x_1, x_2, \cdots, x_n)$ and $HK = (g, h_1, h_2, \cdots, h_n)$;

Algorithm 1: New chameleon hash algorithm.

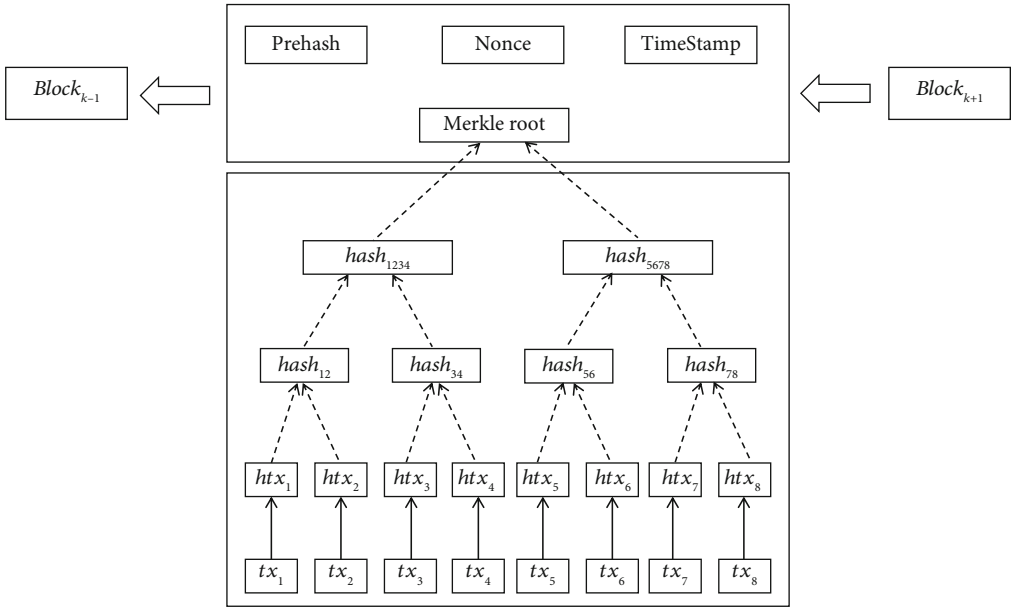

Figure 2: Block structure with new chameleon hash.

(2) The tenant submits their payment address to the smart contract server and the tenant deposits a certain amount of RentCoin to the payment address hinging on the rent

(3) The landlord and tenant create a specific lease contract for the housing lease and then send the finalized lease contract to the smart contract server. After the corresponding smart contract is generated, it will be sent to the owner as well as tenant for signature. Furthermore, both parties will use their own trapdoor private keys to sign after confirmation

(4) The signed smart contract is sent to the leader of the consortium blockchain, who verifies that the signature is correct and broadcasts the result to the other accounting nodes. After other nodes sign the contract, it is stored in the block. When triggering the smart contract, the automatic execution starts

(5) The process is shown in Figure 3 below

Before the smart contract starts to execute, the tenant transfers the deposit and the contracted rent in the form of RentCoin to the payment address, assuming that the tenant transfers RentCoin of b to the payment address. After the smart contract starts executing, the deposit and the contracted monthly (or several monthly) rent are transferred from the tenant's payment address to the landlord's payment address, supposing the monthly rent is b_0 and the deposit is b_1. After that the first smart contract implement, the remaining RentCoin in the tenant's payment account is $b - b_0 - b_1$. Each time the smart contract is hit, the lease is checked for expiration. If it has not expired, the rent continues to be transferred from the tenant's account to the landlord's payment account at b_0. If the tenant's lease has not expired and the landlord agrees to the tenant's subletting, the landlord uses his private key signature to authorize the tenant. Subsequently, the tenant can legally sublet after the landlord's signature.

When the lease between the landlord and tenant expires, the smart contract server generates a record to mark the termination of the smart contract.

At the same time, it is published to the consortium blockchain as a transaction and the contract is automatically terminated. Second, if the tenant's payment account is insufficient to pay the next month's rent after the smart contract is executed k times and the lease has not expired, i.e., $b - kb_0 - b_1 < b_0$ also performs the above operation. Therefore, the tenant must ensure that their payment account has

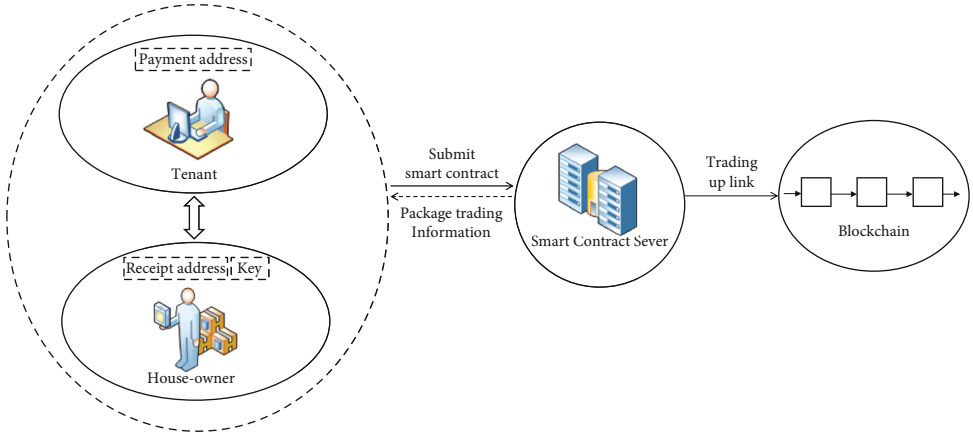

Figure 3: Lease contract generation process.

enough to pay the next month's RentCoin; otherwise, it is treated as a default. The defaulting tenant must pay the landlord a certain amount of default money, which can be deducted from the deposit. Finally, the landlord will return the deposit to the tenant's payment account after the lease expires normally. After the whole rental transaction is completed, the landlord and the tenant have to pay a certain amount of RentCoin to the consortium chain node as transaction fee. If the house has been rented, the status data of the property information update is written to the blockchain.

There is often a shortage of supply during the tenant rental phase, such as a significant increase in the number of rentals during the graduation season. At this time, the number of rented properties will be less than the number of tenants, and there is a noncooperative game among tenants during that period. We observe the healthy development of the housing rental market by solving the Nash equilibrium, through which we can calculate the Nash equilibrium of the housing rental market at different stages. The specific construction and solution process are shown below.

Let $N = 1, 2, \cdots, n$ be the set of tenants, $\forall i \in t$, and let T_i be the set of strategies (the set of listings) of tenant i, which is a nonempty set in R^{k_i}, $T = \prod_{i=1}^{n} T_i$, when tenant i chooses the listings$t_i \in T_i$, $i = 1, 2, \cdots, n$, the benefit that tenant i gets is $f_i(t_1, t_2, \cdots, t_n)$. $\forall i \in N$, let $\hat{i} = N \setminus \{i\}$, $T_{\hat{i}} = \prod_{j \neq i} T_j$, $f_i(t_1, t_2, \cdots, t_n) = f_i(t_i, t_{\hat{i}})$, where $t_{\hat{i}} \in T_{\hat{i}}$. If there exists $t^* = (t_1^*, t_2^*, \cdots, t_n^*) \in T$ such that $\forall i \in N$, with $f_i(t_i^*, t_{\hat{i}}) = \max_{u_i \in T_i} f_i(u_i, t_{\hat{i}}^*)$, then t^* at this point is the Nash equilibrium point of this n-player noncooperative game. At the equilibrium point, each tenant cannot make himself or herself more profitable by individually changing his or her chosen listing.

$\forall i \in N$, let T_i be a nonempty bounded closed convex set in R^{k_i}, $T = \prod_{i=1}^{n} T_i, f_i : T \longrightarrow R$ continuous, and $\forall t_{\hat{i}} \in T_{\hat{i}}, u_i \longrightarrow f_i(u_i, t_{\hat{i}})$ on T_i to be concave, then the Nash equilibrium point of the noncooperative game must exist. This is because, first, $T = \prod_{i=1}^{n} T_i$ must be a nonempty bounded closed convex set in R^k, where $k = k_1 + k_2 + \cdots k_n$. $\forall i \in N, \forall t_{\hat{i}} \in T_{\hat{i}}, F_i(t_{\hat{i}}) = \omega_i \in T_i : f_i(\omega_i, t_{\hat{i}}) = \max_{u_i \in T_i} f_i(u_i, t_{\hat{i}})$.

First, since f_i is continuous when $t_{\hat{i}}$ is fixed, $u_i \longrightarrow f_i(u_i, t_{\hat{i}})$ is continuous, and T_i is a bounded closed set in R^{k_i}, so $F_i(t_{\hat{i}}) \neq \varnothing$. T_i is a boundary, then $F_i(t_{\hat{i}})$ there must be a boundary. Next, note that $\max_{u_i \in T_i} f_i(u_i, t_{\hat{i}}) = c$, $\forall \omega_i^m \in F_i(t_{\hat{i}})$, $\omega_i^m \longrightarrow \omega_i$, then $\omega_i^m \in T_i$. Since T_i is a closed set, $\omega_i \in T_i$. $f_i(\omega_i^m, t_{\hat{i}}) = c$, and since f_i is continuous, $f_i(\omega_i, t_{\hat{i}}) = c, \omega_i \in F_i(t_{\hat{i}})$, $F_i(t_{\hat{i}})$ must be a closed set. $\forall \omega_i^1, \omega_i^2 \in F_i(t_{\hat{i}}), \forall \xi \in (0, 1)$, as $\omega_i^1, \omega_i^2 \in T_i$, T_i is a convex set, $\xi\omega_i^1 + (1 - \xi)\omega_i^2 \in T_i, f_i(\xi\omega_i^1 + (1 - \xi)\omega_i^2) \leq c$. $f_i(\omega_i^1, t_{\hat{i}}) = f_i(\omega_i^2, t_{\hat{i}}) = c$, when $t_{\hat{i}}$ is fixed, $u_i \longrightarrow f_i(u_i, t_{\hat{i}})$ is concave on T_i. So $f_i(\xi\omega_i^1 + (1 - \xi)\omega_i^2, t_{\hat{i}}) \geq \min f_i(\omega_i^1, t_{\hat{i}}) = f_i(\omega_i^2, t_{\hat{i}}) = c$ holds. Therefore, $f_i(\omega_i^1, t_{\hat{i}}) f_i(\xi\omega_i^1 + (1 - \xi)\omega_i^2, t_{\hat{i}}) = c$
$, f_i(\xi\omega_i^1 + (1 - \xi)\omega_i^2 \leq c\xi\omega_i^1 + (1 - \xi)\omega_i^2 \in F_i(t_{\hat{i}}))$, $F_i(t_{\hat{i}})$ must be a convex set. $\forall i \in N$, $F_i(t_{\hat{i}})$ is a nonempty bounded closed convex set in R^{k_i}. Because $F(t) = \prod_{i=1}^{n} F_i(t_{\hat{i}})$, it must be a nonempty bounded closed convex set in R^k, where $k = k_1 + k_2 + \cdots k_n$.

Finally, $\forall i \in N$, since $f_i(u_i, t_{\hat{i}})$ is continuous and T_i is a bounded closed set, $\forall t_{\hat{i}} \in T_{\hat{i}}$ the set-valued map from $G_i(t_{\hat{i}}) = T_i$ must be continuous. T_i is a bounded closed set, and by the great value theorem [28], the extremal map $F_i : T_{\hat{i}} \longrightarrow P_0(T_i)$ must be upper semicontinuous. $\forall t \in T$, since $F(t) = \prod_{i=1}^{n} F_i(t_{\hat{i}})$, best response mapping $F : T \longrightarrow P_0(T)$ on must be upper semicontinuous. Thus, by Kakutani fixed point theorem [29], there exists $t^* \in T$ such that, $t^* \in F(t^*)$, then t^* must be a Nash equilibrium point of the noncooperative game. In other words, if we can find the Nash equilibrium point during the peak rental period, we can prove to a certain extent that the housing rental market is healthy; if we cannot find the Nash equilibrium point, we cannot simply infer that the housing rental market is unhealthy, and we need to consider a number of factors.

5.4. Initiation of Modification Request Phase

(1) If the owner needs to modify the content of the transaction due to policy or rent change, he/she needs to submit a request to the accounting node of the consortium blockchain to change the content

of the historical block m to m'. The owner signs the request with his/her private key sk_{H_i} to get σ_s, and the leader of the consortium chain broadcasts (request, σ_s) to the accounting nodes of the consortium chain that starts the voting phase

(2) After receiving the request, the accounting nodes in the consortium chain sign and broadcast the initiated request if they agree to the modification

(3) After the leader collects signatures from greater than half of the consortium chain's accounting nodes (assuming that the number of signed users is $y > n/2$), he broadcasts these y signatures

5.5. Select the Change Block User Stage

(1) The y consortium chain accounting nodes participating in the voting are noted as $(P_1, P_2, \cdots, P_y)$. Each node P_i chooses a random number ρ_i, and according to (y, n) verifiable secret sharing shares ρ_i to other nodes. The sharing value of ρ_i is recorded as $(s_{i,1}, s_{i,2}, \cdots, s_{i,n})$

(2) Each consortium chain's accounting node P_i verifies the shared values after receiving. The shared values $s_{i,1}, s_{i,2}, \cdots, s_{i,n}$ after passing the verification are summed to $S_i = s_{i,1} + s_{i,2} + \cdots + s_{i,n}$, broadcasting S_i

(3) After each consortium chain accounting node receives at least y of S_i, the value of the random number ρ is calculated by Lagrange interpolation, $\rho = \rho_1 + \rho_2 + \cdots + \rho_y$

(4) Each consortium chain accounting node computes the hash value $h_i = \mathrm{Hash}(\rho, ID_{C_i}), i \in \{1, 2, \cdots, y\}$. The computed y hash values are sorted and the public key ID_{C_s} corresponding to the smallest hash value h_s is selected as the user P_s who modifies the block

5.6. Change the Block Content and Confirm Phase

(1) After selecting the consortium chain accounting node P_s that modifies the block, P_s uses its own private key sk_{C_s} to modify the message m of the historical block to m', and the calculated $r_s' = (m - m' + x_s r_s) x_s^{-1} \bmod q$. The content of the modified block header becomes $(m', (r_1, \cdots, r_s', \cdots, r_n))$, i.e. the rest remains the same except that m and r_s become m' and r_s'

(2) The node P_s will broadcast $(m', (r_1, \cdots, r_s', \cdots, r_n))$, the votes of other users on the modification requests, the obtained random numbers ρ and the signatures on these contents

(3) After the other nodes of the consortium chain receive the broadcast, they verify whether the node P_s that modifies the consortium chain block corresponds to the hash value $\mathrm{Hash}(\rho, ID_{C_s})$ is the smallest, and use the public key ID_{C_s} of the user to verify his signature and the signatures of other users at the voting stage. If all the above verifications pass, then verify the $\mathrm{Hash}(m', (r_1, \cdots, r_s', \cdots, r_n))$ and $\mathrm{Hash}(m, (r_1, \cdots, r_s, \cdots, r_n))$ are equal or not. If all of them are verified, the historical blocks after the changes are recorded and marked with all the information broadcast by P_s. The content of the tag contains all the information broadcast by P_s who modifies the consortium chain block

(4) When using trapdoor keys to modify the contents of historical blocks, signed consent from other nodes needs to be obtained. This approach has somewhat curbed the reckless rent inflation by landlords

6. Conclusion

Compared with the blockchain-based scheme proposed by Lin et al. to build a housing rental ecosystem [30], we introduce chameleon hashing to give each node of the consortium chain the right to modify historical blocks and use n noncooperative game to detect the health of the housing rental market in terms of undersupply to a certain extent. Both schemes apply blockchain technology to the housing rental market and propose solutions to various problems in the housing rental market from different perspectives, respectively.

Our proposal focuses on solving the problems of information asymmetry, illegal subletting, and tenant information leakage in the housing leasing market. The leasing contract based on smart contract enhances the fairness and convenience of transactions in the housing rental market to a certain extent. In the future, we will improve the evaluation and reward mechanism of the scheme as well as enrich the information of rental properties and other issues.

Data Availability

We did not use any external data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 12061020), Foundation of National Natural Science Foundation of China (Grant no. 61962009), Major Scientific and Technological Special Project of Guizhou Province (20183001), Science and Technology Support Plan of Guizhou Province ([2020] 2Y011), and Foundation of Guangxi Key Laboratory of Cryptography and Information Security (GCIS202118).

References

[1] C. Xiaoqing, D. Yao, Z. Liang et al., "The principle and core technology of blockchain," *Chinese Journal of Computers*, vol. 44, no. 1, pp. 84–126, 2021.

[2] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.

[3] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?," *International Journal of Intelligent Systems*, 2021.

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

[5] H. M. Krawczyk and T. D. Rabin, "Chameleon hashing and signatures," 2000, US, US6108783 A.

[6] X. Chen, F. Zhang, W. Susilo, H. Tian, J. Li, and K. Kim, "Identity-based chameleon hashing and signatures without key exposure," *Information Sciences an International Journal*, vol. 265, pp. 198–210, 2014.

[7] S. Nakamotos, "Bitcoin: a peer-to-peer electronic cash system," 2009, https://bitcoin.org/bitcoin.pdf.

[8] J. Kang, R. Yu, X. Huang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.

[9] R. Subramaniam, S. R. Azzuhri, and T. Y. Wah, "Enhanced security approach powered by blockchain technology with NFC to prevent fraudulence in bank letter of credits," in *IECC 2020: 2020 2nd International Electronics Communication Conference*, New York, NY, USA, 2020.

[10] J. Xu, K. Xue, H. Tian, J. Hong, D. S. L. Wei, and P. Hong, "An identity management and authentication scheme based on redactable blockchain for mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6688–6698, 2020.

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

[12] K. Ashritha, M. Sindhu, and K. V. Lakshmy, "Redactable blockchain using enhanced chameleon hash function," in *2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2019.

[13] J. Xu, K. Xue, S. Li et al., "Healthchain: a blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770–8781, 2019.

[14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[15] G. R. Blakley, "Safeguarding cryptographic keys," in *American Federation of Information Processing Societies*, USA, 1979.

[16] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *26th Annual Symposium on Foundations of Computer Science*, Portland, Oregon, USA, 1985.

[17] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proceedings of 28th IEEE Symposium on Foundations of Computer Science*, Los Angeles, CA, USA, 1987.

[18] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology — CRYPTO '91. CRYPTO 1991*, J. Feigenbaum, Ed., vol. 576 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 1991.

[19] M. Benor, S. Goldwasser, and A. Windgerson, "Completeness theorems for noncryptographic fault—tolerant distributed computation," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, Chicago, IL, USA, 1988.

[20] K. C. Toth and A. Anderson-Priddy, "Self-sovereign digital identity: a paradigm shift for identity," *IEEE Security and Privacy Magazine*, vol. 17, no. 3, pp. 17–27, 2019.

[21] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain-or-rewriting history in bitcoin and friends," in *2017 IEEE European symposium on security and privacy*, Paris, France, 2017.

[22] P. Li, H. Xu, T. Ma, and Y. Mu, "Reasearch on fault-correcting blockchain technology," *Journal of Cryptologic Research*, vol. 7, pp. 401–405, 2018.

[23] J. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, pp. 48-49, 1950.

[24] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[25] N. Szabo, "Smart contracts," 1994.

[26] R. Wang, W. T. Tsai, J. He, C. Liu, Q. Li, and E. Deng, "A medical data sharing platform based on permissioned blockchains," in *The 2018 International Conference*, New York, NY, USA, 2018.

[27] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Operating Systems. Design Implementation*, USA, 1999.

[28] C. Berge, *Topological Spaces*, Mac Millan, New York, 1963.

[29] S. Kautani, "A generalization of Brouwer's fixed point theorem," *Duke Mathematical Journal*, vol. 8, no. 3, pp. 457–459, 1941.

[30] Y. Lin, M. Shi, and L. Chen, "Research on the application of blockchain technology in building housing rental information ecosystem," *Price: Theory & Practice*, vol. 10, pp. 56–59, 2020.