蘇品長、夏君和、蘇泰昌 (2022),「題目建構具安全性的智慧合約共享方案-以房屋共享為例」,*資訊管理學報*,第二十九卷,第三期,頁 253-275。

建構具安全性的智慧合約共享方案-以房屋共享為例

蘇品長
國防大學資訊管理學系

夏君和
國防大學資訊管理學系

蘇泰昌*
國防大學資訊管理學系

摘要

共享經濟是透過分享的方式使閒置資源的運用更為便利,顛覆了許多企業與個人的消費模式。惟現行共享經濟平台藉由銀行及可信任第三方進行支付,衍生出許多問題,例如可信任的第三方平台單點故障及缺乏用戶隱私等。以往研究中曾提出透過打破使用者與財產擁有者之間的聯繫來保護隱私,而且確保參與交易的各方之間的公平性,惟未結合智慧合約的自動化流程管理,設計從部署至交易過程條件觸發之交易協定,進一步提升交易之安全性及使用者之隱私性。因此,本研究之目的,在於設計一套基於智慧合約的房屋共享方案。其具體貢獻包括:(1)藉由自我認證機制來強化安全性。(2)透過合約函式自動解決交易衝突,以降低第三方之參與。(3)於交易階段使用盲簽密技術強化個人資訊保密,以保障交易隱私性。

關鍵詞:房屋共享、智能合約、自我認證、盲簽密

* 本文通訊作者。電子郵件信箱:believe50405@gmail.com
2021/12/23 投稿;2022/03/06 修訂;2022/06/07 接受

Su, P.C., Xia, J.H. & Su, T.C. (2022). Design of Secure Sharing Scheme Based on Smart Contracts: Case Study on Home Sharing. *Journal of Information Management*, 29(3), 253-275.

Design of Secure Sharing Scheme Based on Smart Contracts: Case Study on Home Sharing

Pin-Chang Su
Department of Information Management, National Defense University

Jyun-He Xia
Department of Information Management, National Defense University

Tai-Chang Su*
Department of Information Management, National Defense University

Abstract

The sharing economy is a system in which fixed resources are shared to increase their utility, and has disrupted a number of consumer markets for both enterprises and individuals. However, because payments for services on a sharing economy platform are usually conducted by banks and trusted third parties, these third-party platforms may become a single point of failure or compromise user privacy. In a previous study, a privacy mechanism that disrupts the contact between users and asset owners was proposed, which ensures fairness between the participants of a transaction. However, it does not use the automation provided by smart contracts in the design of conditionally triggered transactional agreements for each phase of a sharing-economy transaction (i.e., the initialization, registration, matching, and transaction phases), which would further enhance the transactional security and user privacy. To address this shortcoming, we designed a smart contract-based home sharing scheme. The features of this scheme are as follows: (1) A self-certification mechanism is used to enhance the level of security. (2) Contract functions are used to automatically resolve transactional conflicts and minimize the participation from third parties. (3) Blind signing is used in the transaction stage to preserve the confidentiality of personal information, and thus ensure the privacy of the transaction.

Keywords: Home Sharing, Smart Contracts, Self-certification, Blind Signing

* Corresponding author. Email: believe50405@gmail.com

2021/12/23 received; 2022/03/06 revised; 2022/06/07 accepted

1. INTRODUCTION

The sharing economy has disrupted the consumption patterns of both enterprises and private consumers in a variety of markets. Regardless of whether it is applied to houses or vehicle rentals, the reuse of idle assets, or the sale of intangible assets and skills, the sharing model always ensures that resources are utilized as effectively and efficiently as possible (Chiu, 2014). Airbnb for example, allows renters (tenants) to book accommodation in private housing. This service is now available in 34,000 cities spanning 192 countries, and has over 2 million listings and 60 million guests (Constantiou et al. 2017). The trust mechanism used by Airbnb can be divided roughly into two levels: The first level is the *a priori* exposure of information. Airbnb requires its users to register with their real name and phone number and encourages the users to validate the veracity of this information. They also make this information public, thereby eliminating worries about anonymous users. The second level is the *post hoc* review mechanism, which allows users to write reviews about their experience with a host. A host that receives high review scores will gain more exposure on the Airbnb platform. In short, the Airbnb platform reduces information asymmetry and transactional risks through these information exposures/validations and the aforementioned information exchange mechanism. However, the exposure of personal information will inevitably lead to a loss of privacy and rent discrimination (王文宇, 2018). According to Niya et al. (2018), the problem of sharing economy platforms is that they are excessively reliant on trusted third parties (TTPs), which is often disadvantageous for consumers. Consumers must register separately on every platform, and in most cases they are mandated to give their private data to the platform.

In summary, sharing economy platforms lack user privacy and are susceptible to having the TTP become a single point of failure (SPOF). Fan and Zhang (2019) also noted that centralized data storage methods are often susceptible to malicious data tampering and are likely to incur a SPOF. To enhance trust between lessors and tenants without compromising user privacy, Lee (2019) proposed an automated smart contract-based mechanism that minimizes contractual conflicts or errors. Likewise, Yuan and Wang (2016) presented a method based on blockchain nodes to address problems such as high transactional costs, low efficiency, and unsafe data storage, which are common in centralized architectures. However, although the interaction through the blockchain wallet address is anonymous, it doesn’t operate under the external supervision mechanism, and the mediation of the transaction process in the sharing mechanism still requires the participation of some third parties. Thus, we will attempt to address the aforementioned problems by using a smart contract proposed

by Karamitsos, Maria, and Al Barghuthi, (2018) for real estate transactions, an Android-based peer-to-peer purchase and rental application designed by Niya et al. (2018), and service level agreements proposed by Hang and Kim for enhanced decentralized sharing economy services. The procedures of our transaction agreements were designed according to the findings of these studies. We will also utilize the fast and privacy-preserving method based on permissioned blockchain (FPPB) method proposed by Li et al. (2018), which uses stealth addresses and zero-knowledge proofs on a permissioned blockchain to ensure privacy, uniformity of the transaction contents, and fairness in all sharing economy transactions. In addition, we incorporated the blind signature mechanism of Liu et al. (2018) to ensure fairness between all transaction participants, as well as the automated smart contract-based payment scheme of Yu (2020).

1.1 Contribution

We have constructed a scheme that uses a smart contract to conditionally trigger transactional agreements for each phase of a sharing-economy transaction (i.e., the initialization, registration, matching, and transaction phases). A self-certification mechanism is also used in this system, and thus the hosts and tenants will be able to verify the identities of other individuals without using a third party. The blind signature technique further enhances the security and privacy of all transactions on our scheme. Therefore, the proposed protocol significantly shortens the transaction process, and reduces the time costs of the protocol operation.

1.2 Limitation

This study focuses on logic and algorithmic derivation, so we don't develop the program and perform system simulation. Because of the decentralized nature of this PKI, there is no single authority that can maintain a local dictionary data structure for efficient public key lookup. We therefore separate the functionality of verifying a known public key from that of looking up a new public key, and leverage secures distributed data structures to support each of them efficiently (Fromknecht et al. 2014). Thus, we can prove our scheme does not incur administrative overhead due to the use of a large number of keys for authentication and can guarantee better system performance.

In our research, we know that a smart contract is an automatically executed program triggered by the corresponding inputs, and a smart contract is publicly accessible. Our scheme will interact with the user through the public and private keys of the smart contract provider, then the data is sent to the blockchain by the smart contract.

2. LITERATURE REVIEW

Based on recent studies, we proposed a blockchain-based smart contract payment system. The scope of this review includes studies on the development of the sharing economy and sharing-economy platforms, blockchain technology, and the blind signature encryption technique.

2.1 Sharing Economy

The sharing economy (also known as collaborative consumption) is an emerging, rapidly developing mode of commerce. In the last decade, many forms of collaboration have emerged in the business world contributing to new and different systems of commerce (i.e, Airbnb, Uber, Zipcar), even the value proposition of agriculture social enterprises with community supported agriculture nearly alike value proposition of CC and sharing economy (Tung & Chiu ,2019). Botsman and Rogers (2010) divided the sharing economy into three types: (1) product-service systems, which allow pre-existing resources to be rented to others, (2) redistribution markets, which move used items from places where they are not needed to persons or places where they are, and (3) collaborative life-cycles, which gather people with similar needs or interests to share/exchange latent resources such as time, space, and skills to maximize the efficiency of these resources. Airbnb is one of the world's largest collections of unique travel accommodations and experiences, and its hosts have created over 7 million accommodation options and 50,000 experiences (Airbnb, 2020). Airbnb's business model is to allow potential hosts (lessors) to list their spare rooms as guest accommodations, and thus charge nightly, weekly, or monthly rentals. Airbnb then takes a commission from the resulting rental fees (9% to 12%), depending on the length of the stay. The lessor also pays 3% as a payment processing fee. Airbnb also constructed an online trust system, which allows all participants to review their lodgings or tenants and provides rewards to the best performers in this regard.

2.2 Smart Contracts

In 2008, Nakamoto published Bitcoin: A Peer-to-Peer Electronic Cash System, which described Bitcoin and its algorithms, and made the first known mention of blockchain technology. A blockchain may be defined as a database that is fully shared between all of its users, which enables the trade of valuable assets without relying on intermediaries or a centralized mechanism (Risius and Spohrer, 2017). The idea of smart contracts was first proposed by Nick Szabo in 1994 (Szabo, 1994), and their purpose is to digitally transmit, formulate, and validate computer contracts, that is, to serve as electronic contracts. In general, the objective of a smart contract is to satisfy common contractual conditions (e.g., payment terms, liens, confidentiality, and enforcement) and minimize exceptions. It may therefore be stated that a smart

contract uses programmatic logic to implement the terms and conditions of a transactional contract. Although smart contracts differ from one blockchain platform to another, they generally operate in an "event-driven" manner (Chen, 2018).

2.3 Blind Signatures

Blind signing is a technique that allows a signature requester to have a signer sign a message without knowing its contents, thus preventing any leakage of information. To this end, the signature requester will first combine the document with some "blinding factor," and send the resulting blinded document to the signer. The signer then signs the blinded document using their private key and returns it to the signature requester. The signature requester will then de-blind the document and finally obtain the true digital signature (Chaum, 1984). Because digital signatures were previously unable to ensure the confidentiality of transmitted documents, Zheng (1997) proposed a technique called signcryption, which combines the discrete logarithm problem (DLP)-based digital signatures with hybrid encryption. In signcryption, the document is first signed. The ciphertext is then produced by applying symmetric-key encryption on the cleartext, using keys that were generated by the sender and recipient using the signcryption mechanism (Lai, 2003).

3. METHOD AND FRAMEWORK

In this study, we designed a fair and anonymous online transaction protocol based on Ethereum smart contracts, using insight gleaned from previous studies. Because sharing platforms typically require an identity registration, we used an alliance chain with the certificate authority (CA), management platform, lessors, and tenants as its nodes. Because it uses a virtually tamper-proof distributed ledger of transactions, our transaction protocol is highly trustworthy. The CA is responsible for identity verification and the generation of public and private keys, and is used to register the identities of the management platform, tenants, and lessors, and to provide security functions such as signing, encryption, and verification. The clients will use a digital app (DApp) to call the smart contract (SC) and upload their information to the chain, which is then propagated throughout the blockchain network. Because blockchains have a limited storage capacity, an InterPlanetary File System (IPFS) is used in conjunction with the SC. To provide traceability and authenticity, all personal information, listing information, and user reviews are stored in the distributed IPFS database, whereas their hashes are stored in the SC. The DApp front-end provides logical processing and calls the address of the SC to execute all contractual conditions.

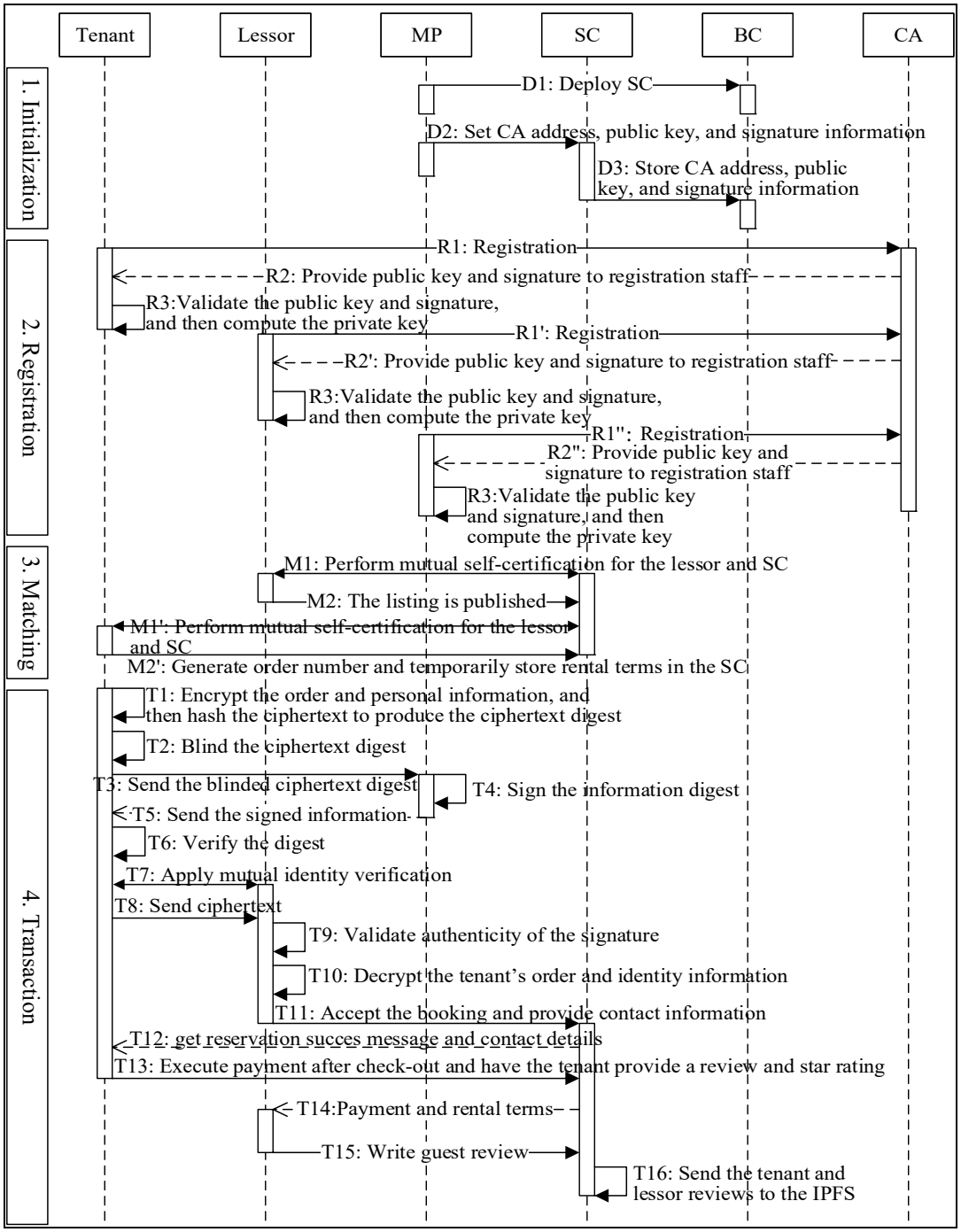
3.1 Framework and Symbols of Transaction Process

The procedures of our system may be divided into the initialization, registration, authorization, and transaction phases. The symbols and parameters used in each stage are described in Table 1.

Table 1: Description of symbols and parameters

No.	Symbols	Description
1	CA, MP, T, L, SC, BC, Ps	The certificate authority (CA), management platform (MP), tenant (T), lessor (L), smart contract (SC), blockchain (BC), and participants (general term for the aforementioned roles) (Ps)
2	P_{Ps}	The verifiable public keys of the participants
3	PK_{Ps}, sk_{Ps}	The public keys and private key of the participants
4	$E(F_q)$	An elliptic curve in the finite domain F_q
5	$h(C)$	The hashing of ciphertext C to convert it into a ciphertext digest
6	$createListing()$	A lessor listing their property on the blockchain
7	$sendBlindSigncryption()$	Send the encrypted and blinded order
8	$returnSignature\{\}$	The SC returning a signature
9	$register()$	Registers a participant with the CA
10	$verify()$	The CA verifies the public key and signature of a participant
11	$calculate()$	A participant calculating their private key using the CA parameters
12	$sendOrder()$	The tenant sending an order
13	$verifyOrder()$	A tenant verifying the signature of an order
14	$confirmOrder()$	The lessor confirms the check-in authorization sent by the SC
15	$paymentNum$	The number of orders that are generated after the tenant has selected a listing
16	$contentHash$	Hash value of the order information
17	$starRating$	Review and star rating given by the tenant and lessor
18	$payRent$	Rent payment by the tenant
19	$checkRent$	Confirmation of rent payment by the lessor
20	$listingInfo$	*listingInfo* is the order generated for the tenant (which includes the number of the listing that the tenant wishes to rent, the date of the rental, and the name and phone number of the tenant)
21	$M,\ c$	Cleartext information and cipher digest
22	$t,\ e$	Random values generated by the tenant and SC

23	Θ	The ciphertext digest c that was blinded by the tenant
24	R, S	The validation value and signature generated by the encrypted and blinded order
25	$S',\ c',\ PK'_T, PK'_L,\ M'$	Calculated validation values, which are compared to S, c, PK_T, PK_L, and M
26	ID_x, d_x, V_x, w_x, k_x	ID_x, Identity information of x; d_x, random secret parameters selected by x; V_x, signature file of x; w_x, signature of x calculated by the CA; k_x, random parameter of the CA



3.2 Phases and Procedures of the Protocol and Its Algorithms

The procedures of our system are expressed in terms of the unified modeling language (UML), and are divided into four phases: the initialization, registration, matching, and transaction phases (as shown in Fig. 1). In the following, we describe the information transfers and algorithms of our protocol in the SC initialization, registration, matching, and transaction phases.

Figure 1: Procedures of each phase of the proposed transaction protocol

1. Initialization

During this phase, the MP deploys the SC on the blockchain network (BC), and then defines the identity information, public key, and certificate information of the CA. This information is then stored on the BC.

$$\mathbf{D1}.\ \boldsymbol{MP} \xrightarrow{\boldsymbol{Deploy}} \boldsymbol{BC}:\boldsymbol{SC} \tag{1}$$

The MP deploys the SC.

$$\mathbf{D2}.\ \boldsymbol{MP} \rightarrow \boldsymbol{SC}:\boldsymbol{modifierMP}\left(\boldsymbol{setCA}\left(\boldsymbol{ID_{CA}}, \boldsymbol{PK_{CA}}, \boldsymbol{Cert_{CA}}\right)\right) \tag{2}$$

The MP sets the identity information ($\boldsymbol{ID_{CA}}$), public key ($\boldsymbol{PK_{CA}}$), and certificate information ($\boldsymbol{Cert_{CA}}$) of the CA.

$$\mathbf{D3}.\ \boldsymbol{SC} \rightarrow \boldsymbol{BC}:\boldsymbol{ID_{CA}}, \boldsymbol{PK_{CA}}, \boldsymbol{Cert_{CA}} \tag{3}$$

The SC stores the $\boldsymbol{ID_{CA}}$, $\boldsymbol{PK_{CA}}$, and $\boldsymbol{Cert_{CA}}$ information of the CA on the BC.

2. Registration phase

The participants' Ps (tenant T, lessor L, and the SC) provide their personal information to the CA and are thus registered by the CA. A self-certification mechanism is then used to generate public and private key pairs. Finally, certificates are generated using the private key of the CA and the public keys and digital signatures of the participants. In the future, the CA will check whether a participant has already been registered, and the registration step is skipped if the participant has already registered.

The CA will select a secure elliptic curve in the finite domain *Fq*, *E(Fq)*, where *q* is a prime larger than 256 bits. A base point *G* of order n will then be selected on *E(Fq)*, such that $n \cdot G = O$, with *O* being an elliptic curve-point at infinity. A one-way hash function *h( )* and private key $\boldsymbol{sk_{CA}}$ are used to calculate the public key PCA and the parameters of the open system. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{PK_{CA}} = \boldsymbol{sk_{CA}} \cdot \boldsymbol{G} \tag{4}$$

$$\mathbf{R1}.\ \mathbf{T} \rightarrow \mathbf{CA}: \mathbf{register}(\mathbf{ID_T}, \mathbf{V_T}) \tag{5}$$

Likewise,

$$\mathbf{R1}'.\ \boldsymbol{L} \rightarrow \boldsymbol{CA}: \boldsymbol{register}(\boldsymbol{ID_L}, \boldsymbol{V_L}) \tag{6}$$

and

$$\mathbf{R1}''.\ \boldsymbol{SC} \rightarrow \boldsymbol{CA}: \boldsymbol{register}(\boldsymbol{ID_{SC}}, \boldsymbol{V_{SC}}) \tag{7}$$

The participants *T*, *L*, and SC then generate their signatures V_{Ps} by applying a one-way collision-free hash function on their identities ID_{Ps} and random secret parameter d_{Ps} (where $\boldsymbol{d_{Ps} \in [2, n-2]}$). The values of ID_{Ps} and V_{Ps} are then sent to the CA, and the CA will place the registered participants on the blockchain. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{V_{Ps} = h(d_{Ps} \parallel ID_{Ps}) \cdot G} \tag{8}$$

$$\mathbf{R2.}\ \boldsymbol{CA \rightarrow T: verify(P_T, w_T)} \tag{9}$$

Likewise,

$$\mathbf{R2'.}\ \boldsymbol{CA \rightarrow L: verify(P_L, w_L)} \tag{10}$$

and

$$\mathbf{R2''.}\ \boldsymbol{CA \rightarrow SC: verify(P_{SC}, w_{SC})} \tag{11}$$

The CA will select a random value $\boldsymbol{k_{Ps} \in [2, n-2]}$ to calculate the verifiable public key P_{Ps} and signature $\boldsymbol{w_{Ps}}$ of participant *Ps*, and then send it to *Ps*. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{P_{Ps} = V_{Ps} + \left(k_{Ps} - h(ID_{Ps})\right) \cdot G = \left(q_{Psx}, q_{Psy}\right)} \tag{12}$$

$$\boldsymbol{w_{Ps} = k_{Ps} + sk_{CA}\left(q_{Psx} + h(ID_{Ps})\right)} \tag{13}$$

$$\mathbf{R3.}\ \mathbf{T:}\ \boldsymbol{calculate(sk_T)} \tag{14}$$

Similarly,

$$\mathbf{R3'.}\ \boldsymbol{L: calculate(sk_L)} \tag{15}$$

The participants *Ps* compute their secret keys $\boldsymbol{sk_{Ps}}$ using the parameters returned by the CA ($\boldsymbol{w_{Ps}}$ and $\boldsymbol{P_{Ps}}$), as shown below:

$$\boldsymbol{sk_{Ps} = [w_{Ps} + h(d_{Ps} \parallel ID_{Ps})]} \tag{16}$$

And the calculation process of *Ps's* public key $\boldsymbol{PK_{Ps}}$ could verify the verifiable public keys $\boldsymbol{P_{Ps}}$ sent by CA. The proofs of the equations that were incorporated into the ECC are as follows:

$$\boldsymbol{PK_{Ps} = sk_{Ps} \cdot G = [w_{Ps} + h(d_{Ps} \parallel ID_{Ps})] \cdot G} \tag{17}$$

$$\boldsymbol{\because w_{Ps} = k_{Ps} + sk_{CA}\left(q_{Psx} + h(ID_{Ps})\right)} \tag{18}$$

$$\boldsymbol{\therefore PK_{Ps} = \left[k_{Ps} + sk_{CA}\left(q_{Psx} + h(ID_{Ps})\right) + h(d_{Ps} \parallel ID_{Ps})\right] \cdot G} \tag{19}$$

$$\text{then } \boldsymbol{PK_{Ps} = k_{Ps} \cdot G + sk_{CA}\left(q_{Psx} + h(ID_{Ps})\right) \cdot G + [h(d_{Ps} \parallel ID_{Ps})] \cdot G} \tag{20}$$

$$\boldsymbol{\because PK_{CA} = sk_{CA} \cdot G} \tag{21}$$

$$\boldsymbol{\therefore PK_{Ps} = k_{Ps} \cdot G + \left[\left(q_{Psx} + h(ID_{Ps})\right)\right] \cdot PK_{CA} + h(d_{Ps} \parallel ID_{Ps}) \cdot G} \tag{22}$$

$$\boldsymbol{\because V_{Ps} = h(d_{Ps} \parallel ID_{Ps}) \cdot G} \tag{23}$$

$$\boldsymbol{\therefore PK_{Ps} = k_{Ps} \cdot G + \left[\left(q_{Psx} + h(ID_{Ps})\right)\right] \cdot PK_{CA} + V_{Ps}} \tag{24}$$

$$\boldsymbol{\because V_{Ps} = P_{Ps} - \left(k_{Ps} - h(ID_{Ps})\right) \cdot G} \tag{25}$$

$$\begin{aligned}\boldsymbol{\therefore PK_{Ps}} &= \boldsymbol{k_{Ps} \cdot G + \left[\left(q_{Psx} + h(ID_{Ps})\right)\right] \cdot PK_{CA} + P_{Ps} - \left(k_{Ps} - h(ID_{Ps})\right) \cdot G} \\ &= \boldsymbol{k_{Ps} \cdot G + \left[\left(q_{Psx} + h(ID_{Ps})\right)\right] \cdot PK_{CA} + P_{Ps} - k_{Ps} \cdot G + h(ID_{Ps}) \cdot G} \\ &= \boldsymbol{P_{Ps} + h(ID_{Ps}) \cdot G + \left[\left(q_{Psx} + h(ID_{Ps})\right)\right] \cdot PK_{CA}}\end{aligned} \tag{26}$$

$$\text{then } \boldsymbol{P_{Ps} = PK_{Ps} - h(ID_{Ps}) \cdot G - \left[\left(q_{Psx} + h(ID_{Ps})\right)\right] \cdot PK_{CA}} \tag{27}$$

$$\text{if } \boldsymbol{P_{Ps}} = \boldsymbol{P_{Ps}} \text{ sent by CA} \tag{28}$$

$$\text{then Ps's secret key } \boldsymbol{sk_{Ps}} \text{ and public key } \boldsymbol{PK_{Ps}} \text{ are correct} \tag{29}$$

According to these proofs, once the participants have obtained P_{Ps} and w_{Ps} from the CA (by registering their ID_{Ps} with the CA), they could calculate secret key and public key by themselves, and they will be able to validate the correctness of the self-generated PK_{Ps} public keys and thus verify the identities of others, without certification from the CA.

3. Matching phase

In this phase, *L* will call the SC to perform mutual self-certification, and then publish his/her listing on the BC. Next, using the SC, *T* will perform mutual self-certification, and generate an order number. The payment is then temporarily stored on the BC through SC.

$$\mathbf{M1}.\ \boldsymbol{L \leftrightarrow SC: ID_L,\ P_L,\ PK_L} \tag{30}$$

Similarly,

$$\mathbf{M1'}.\ \boldsymbol{T \leftrightarrow SC: ID_T,\ P_T,\ PK_T} \tag{31}$$

Now, *L* and *T* will validate their identity information (ID_{Ps}, P_{Ps}, and PK_{Ps}) with the SC.

$$\boldsymbol{PK'_L = P_L + h(ID_L) \cdot G + \left(q_{Lx} + h(ID_L)\right) \cdot PK_{CA}} \tag{32}$$

$$\boldsymbol{PK'_T = P_T + h(ID_T) \cdot G + \left(q_{Tx} + h(ID_T)\right) \cdot PK_{CA}} \tag{33}$$

$$\mathbf{M2}.\ \boldsymbol{L \xrightarrow{DApp} SC: modifierLL\left(createListing(ListingInfo)\right)} \tag{34}$$

Next, *L* will generate the listing information (this function is limited to successfully registered lessors).

$$\mathbf{M2'}.\ \boldsymbol{T \xrightarrow{DApp} SC: modifierTen(paymentNum, ID_T,\ PK_T, P_T)} \tag{35}$$

Then, *T* will search for published listings (this function is limited to successfully registered tenants) and thus generate an order number. The payment is then temporarily stored on the SC.

4. Transaction phase

The order information that will be sent by *T* will first be encrypted and blinded, and then used to generate a ciphertext digest. This ciphertext digest is then sent to the SC to be signed. The signed digest is returned to *T* and verify the digest. The process of all information about the transaction will be openly stored on the blockchain. Next, *T* and *L* will mutually verify each other and construct a shared secret key, *Key*. The ciphertext is then hashed and sent to *L*, which will subsequently verify the correctness of the SC's signature, and then decrypt the ciphertext to obtain the order information. If the order is accepted, the SC will be triggered to make the check-in a permissible event and stored on the blockchain.

The SC will then check in the authorized tenant *T*. After *T* has checked out, the payment function will be triggered, and *T* will be allowed to review and rate *L*. Finally, the payment will be sent by the SC to *L*, and *L* may write a review for *T* after receiving the payment. The SC will then upload the review and star ratings onto the IPFS.

The public parameters of the system are an elliptic curve *E* and its modulus *q*. In addition, *T* will choose some integer sk_T ($0 < sk_T < p$) and some $\boldsymbol{G} \in \boldsymbol{E}$ to calculate $\boldsymbol{PK_L} = \boldsymbol{sk_T} \cdot \boldsymbol{G}$. The public key is then $\boldsymbol{G}, \boldsymbol{PK_T}$, whereas the private key is sk_L.

T1. Encryption procedure

Given a cleartext M = (m_1, m_2), let there be some *G* that is not necessarily a point on *E* (and is a cyclic subgroup of *E*), and some number $\boldsymbol{t} \in \boldsymbol{Z_q}$. The ciphertext {$C_1$, C_2} may then be computed as follows:

$$\boldsymbol{C_1} = (\boldsymbol{c_{11}}, \boldsymbol{c_{12}}) = \boldsymbol{t} \cdot \boldsymbol{G} \tag{36}$$

$$\boldsymbol{Y} = (\boldsymbol{y_1}, \boldsymbol{y_2}) = \boldsymbol{t} \cdot \boldsymbol{PK_L} \tag{37}$$

$$\boldsymbol{C_2} = (\boldsymbol{c_{21}}, \boldsymbol{c_{22}}) = \boldsymbol{y_1} \times \boldsymbol{m_1}, \ \boldsymbol{y_2} \times \boldsymbol{m_2} \tag{38}$$

Let $C = \{C_1, C_2\}$

$$\boldsymbol{h}(\boldsymbol{C}) = \text{c} \tag{39}$$

T2. Blinding procedure

Here, *T* will blind the ciphertext digest *c* using a blinding factor ($sk_T \cdot PK_T$) to generate Θ. The equation that was incorporated into the ECC is as follows:

$$\Theta = c \cdot (sk_T \cdot PK_T) \tag{40}$$

$$\text{T3.}\boldsymbol{modifierTen}(\boldsymbol{sendBlindSigncryption}(\boldsymbol{\Theta})) \tag{41}$$

Next, *T* sends the blinded ciphertext digest Θ to SC, and stores it on the blockchain.

T4. Signing phase

After the MP get Θ from blockchain, it will randomly select a second blinding factor $\boldsymbol{e} \in \boldsymbol{Z_q}$ and generate the validation value and signature $\{\boldsymbol{R}, \boldsymbol{S}\}$. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{R} = \boldsymbol{e} \cdot \boldsymbol{\Theta} \tag{42}$$

$$\boldsymbol{S} = (\boldsymbol{e} + \boldsymbol{sk_{MP}}) \cdot \boldsymbol{\Theta} \tag{43}$$

$$\text{T5.}\boldsymbol{MP} \rightarrow \boldsymbol{T}: \boldsymbol{returnSignature}\{\boldsymbol{\Theta}, \boldsymbol{R}, \boldsymbol{S}\} \tag{44}$$

The MP puts it on the blockchain by SC and returns the signature information to *T*, who then stores (Θ, *e*).

T6. Verify the digest phase

After T get $\{\Theta, R, S\}$ from blockchain, T will use its sk_T to decrypt the signature $\{R, S\}$ and use the public key P_{MP} to verify the digest S'. Finally, T calculates c'. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{S' = S - c \cdot sk_T \cdot P_{MP}} \tag{45}$$

$$\boldsymbol{c' = sk_T \cdot (sk_T - 1) \cdot c + c} \tag{46}$$

T7. $$\boldsymbol{T \leftrightarrow L : ID_T,\ PK_T,\ P_T \ \leftrightarrow\ ID_L,\ PK_L,\ P_L} \tag{47}$$

Next, T and L will verify the identities of the other (ID_T, P_T, and PK_T from T and ID_L, P_L, and PK_L from L). L will check whether PK'_T is a match with PK_T, whereas T will check whether PK'_L is a match with PK_L. After T and L have verified the identity of the other, they will construct the shared key $Key_{(L,T)}$. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{PK'_T = P_T + h(ID_T) \cdot G + \left(q_{Tx} + h(ID_T)\right) \cdot PK_{CA}} \tag{48}$$

$$\boldsymbol{PK'_L = P_L + h(ID_L) \cdot G + \left(q_{Lx} + h(ID_L)\right) \cdot PK_{CA}} \tag{49}$$

$$\boldsymbol{Key_{(L,T)} = sk_L \cdot PK_T} \tag{50}$$

T8. $$\boldsymbol{T \rightarrow L: sendOrder\ \{c',\ S',\ R,\ C\}} \tag{51}$$

Here, T will send the verified digest and signed order information $\{c', S', R, C\}$ to L.

T9.L: $$\boldsymbol{verifyOrder\ \{\acute{c},\ \acute{S},\ R,\ C\}} \tag{52}$$

In addition, L will use PK_{MP} to validate the signature on $\{c', S', R, C\}$ (the order information sent by T), to establish whether $\boldsymbol{R - h(C) \cdot PK_{MP} \stackrel{?}{=} S' - c' \cdot PK_{MP}}$ holds. The equations that are incorporated into the ECC are as follows:

Left hand side:

$$\boldsymbol{R - h(C) \cdot PK_{MP} = e \cdot c \cdot\ sk_T \cdot PK_T - c \cdot PK_{MP}} \tag{53}$$

Right hand side:

$$\begin{aligned}
\boldsymbol{S' - c' \cdot PK_{MP}} &= \boldsymbol{S - c \cdot\ sk_T \cdot PK_{MP} - [sk_T \cdot (sk_T - 1) \cdot c + c] \cdot PK_{MP}} \\
&= \boldsymbol{(sk_{MP} + e) \cdot c \cdot\ sk_T \cdot PK_T - c \cdot\ sk_T \cdot PK_{MP} - [sk_T \cdot (sk_T - 1) \cdot c + c] \cdot PK_{MP}} \\
&= \boldsymbol{(sk_{MP} + e) \cdot c \cdot\ sk_T \cdot PK_T - c \cdot\ sk_T \cdot PK_{MP} - [sk_T \cdot sk_T \cdot c - sk_T \cdot c + c] \cdot PK_{MP}} \\
&= \boldsymbol{(sk_{MP} + e) \cdot c \cdot\ sk_T \cdot PK_T - c \cdot\ sk_T \cdot PK_{MP} - [sk_T \cdot sk_T \cdot c - sk_T \cdot c + c] \cdot PK_{MP}} \\
&= \boldsymbol{(sk_{MP} + e) \cdot c \cdot\ sk_T \cdot PK_T - c \cdot\ sk_T \cdot P_{MP} - sk_T^2 \cdot c \cdot P_{MP} + sk_T \cdot c \cdot P_{MP} - c \cdot PK_{MP}} \\
&= \boldsymbol{sk_{MP} \cdot c \cdot sk_T \cdot PK_T + e \cdot c \cdot sk_T \cdot PK_T - sk_T^2 \cdot c \cdot PK_{MP} - c \cdot PK_{MP}} \\
&= \boldsymbol{e \cdot c \cdot sk_T \cdot PK_T - c \cdot PK_{MP}}
\end{aligned} \tag{54}$$

T10. Decryption by L

Here, L uses sk_L to decrypt C_1 and calculate $Z = (z_1, z_2)$. The equation incorporated into the ECC is

$$\boldsymbol{Z = (z_1,\ z_2) = sk_L \cdot C_1 = sk_L \cdot t \cdot G = t \cdot PK_L}\text{I}. \quad (55)$$

Then, L uses the inverse element of point Z and c_2 to obtain M'. Finally, the value of c' that was obtained in T9 is checked against $h(C)$; if these values are the same, L has then obtained the cleartext M. The equations that are incorporated into the ECC are as follows:

$$\boldsymbol{M' = (c_{21} \cdot z_1^{-1},\ c_{22} \cdot z_2^{-1}) = (y_1 \cdot m_1 \cdot z_1^{-1},\ y_2 \cdot m_2 \cdot z_2^{-1}) = (m_1',\ m_2')} \quad (56)$$

$$\boldsymbol{h(C) \stackrel{?}{=} c'} \quad (57)$$

$$\boldsymbol{M' = M} \quad (58)$$

$$\text{T11.}\boldsymbol{L \rightarrow SC: modifierLL(comfirmOrder)} \quad (59)$$

After L receives the order, the SC is triggered to make the check-in into the listing a permissible event.

$$\text{T12.}\boldsymbol{SC \rightarrow T: modifierTen(confirmOrder(true), contentHash)} \quad (60)$$

The SC transfers the permission to check-in to T.

$$\text{T13.}\ \boldsymbol{T \rightarrow SC: modifierTen(finish(true), starRating)} \quad (61)$$

Here, T pays after checking out and gives L a star rating and review.

$$\text{T14.}\boldsymbol{SC \rightarrow L: payRent} \quad (62)$$

The SC sends the rent payment to L.

$$\text{T15.}\ \boldsymbol{L \rightarrow SC: modifierLL(chekRent(true), starRating)} \quad (63)$$

Then, L receives the rent payment and writes a review for T.

$$\text{T16.}\boldsymbol{SC \rightarrow IPFS: starRating(T, L)} \quad (64)$$

The ratings and reviews of T and L are transferred by the SC to the IPFS.

4. SECURITY ANALYSIS AND ASSESSMENT

By using blind signatures, smart contracts, and blockchain technology, we have designed a home-sharing protocol that is fair, private, and resistant to double-spending attacks, which also minimizes third-party participation. The tenant will first blind and encrypt their order, which is signed by the SC. The signed order is then verified the digest by the tenant before being sent to the lessor, who will decrypt and confirm the order. The lessor then sends his/her contact details to the tenant through the SC. Our SC-based homesharing scheme satisfies the requirements of the National Institute of Standards and Technology (NIST) in terms of the minimization of third-party participation, fairness, and privacy (Yaga et al., 2018; Lesavre et al., 2021), and thus ensures the security of all participants and transactions on our system. Our scheme also conforms to the information security requirements of the ISO (2005) standard,

which include confidentiality, integrity, authenticity, and non-repudiation. Our system also provides anonymity and unforgeability through the use of blind signatures and has a self-certification mechanism. In the following, we will analyze the security of the proposed scheme based on the aforementioned metrics, that is, by defining these metrics, the relevant security scenarios and the solutions to such scenarios are analyzed.

4.1 Minimization of Third-Party Participation

1. Definition: In an online transaction, third-party payment refers to the use of a neutral payment platform to collect and provide payment to the buyer and seller (沈淑惠, 2014).
2. Scenario: During the transaction period, it is inappropriate to allow too many actors to participate in the transaction. Therefore, during the transaction phase, *L* and *T* must be allowed to transact directly with each other, and the MP and CA are forbidden from participating in the transaction.
3. Solution: During the transaction phase, after *T* and *L* have constructed shared keys with the SC, they will be able send their identity verification information to each other during T7 (Equation (44), $T \leftrightarrow L: ID_T,\ PK_T,\ P_T \leftrightarrow ID_L,\ PK_L,\ P_L$) and verify each other (Equations (48) and (49), $PK'_T = P_T + h(ID_T) \cdot G + \left(q_{Tx} + h(ID_T)\right) \cdot P_{CA}$ and $PK'_L = P_L + h(ID_L) \cdot G + \left(q_{Lx} + h(ID_L)\right) \cdot P_{CA}$). Finally, *T* and *L* will construct a shared key that will be used for the transmission of order information (Equation (50), $Key_{(L,T)} = sk_L \cdot PK_T$). After the SC has generated an order for *T*, the *sendOrder*() and *checkOrder*() commands of the transaction phase will be limited to *T* and *L* through *modifierTen*() and *modifierLL*(). This effectively eliminates a third-party participation.

4.2 Fairness

1. Definition: A transaction is fair if both parties of the transaction received the item they expected. Owing to the irreversibility of blockchain technology and the automation and anonymity provided by smart contracts, there are significant limitations in the recovery phase of the transaction. Therefore, fraud prevention and detection must be prioritized by such a protocol (Asokan, 1998).
2. Scenario: Transaction abnormalities occur, such as *T* having insufficient funds to pay a rental fee, or a check-in authorization not being sent to *T* within the allotted time after *L* has received an order.
3. Solution: During the transaction, the coding of the SC will conditionally constrain the payment information sent by *T*. Therefore, *T* will pay when the order number is generated, in step M2' (Equation (38)),

$T \xrightarrow{Dapp} SC : modifierTen(paymentNum, ID_T,\ PK_T, P_T)$. If *T* does not have sufficient funds for the payment, the order will not be confirmed, thus preventing further problems. After *L* receives the order information of *T*, *L* must send its listing information and contact details to *T*. If *T* does not receive a check-in authorization within the allotted time, an error occurs during the integrity check, or if *L* refuses to accept the order, the payment will then be returned to *T*. The integrity check is shown in T12 (Equation (60), $SC \rightarrow T: modifierT(confirmOrder(true),\ contentHash)$).

4.3 Privacy

1. Definition: If an encryption is not applied or a weak encryption is used, it is likely that passwords will be cracked and sensitive information will be leaked (OWASP, 2017).
2. Scenario: A hacker wishes to use an order generated by *T* to steal or misappropriate personal information for illegal purposes.
3. Solution: After *T* has booked a listing on the blockchain and generated the corresponding order information, the order will be blinded (Equation (43), $\Theta = c \cdot (sk_T \cdot PK_T)$), and the blinded ciphertext digest will be sent to the SC to be signed, according to step T3 (Equation (41), $modifierTen(sendBlindSigncryption(\Theta))$). Then, *T* will verify the signed digest that was returned by the SC (Step T8, Equation (51), $T \rightarrow L: sendOrder\ \{c',\ S',\ R,\ C\}$) and send it to *L*. If the information passes the verification (Equations (3)–(53), $R - h(C) \times P_{SC} = e \cdot \mathrm{c} \cdot\ sk_T \cdot PK_T - c \cdot P_{SC}$), *L* will decrypt the order (Equations (3)–(56), $M' = (c_{21} \cdot z_1^{-1},\ c_{22} \cdot z_2^{-1}) = (y_1 \cdot m_1 \cdot z_1^{-1},\ y_2 \cdot m_2 \cdot z_2^{-1}) = (m_1',\ m_2')$) and obtain the cleartext information (Equation (58), *M'* = *M*). This procedure will verify all participant identities and prevent the blockchain from recording any personal information, which ensures the privacy of *T*.

4.4 Confidentiality

1. Definition: In the context of a data transmission or transaction, confidentiality ensures that only authorized persons or programs will be able to obtain information regarding the data or transaction, thus preventing data leakage.
2. Scenario: A hacker attempts to intercept information about a transaction between *T* and *L*.
3. Solution: When *T* and *L* are in the midst of a transaction, they will verify the identity of the other through step T7 (Equation (47), $SC \leftrightarrow L: ID_T,\ PK_T,\ P_T \leftrightarrow ID_L,\ PK_L,\ P_L$) and then generate a shared key (Equation (50),

$Key_{(L,T)} = sk_L \cdot PK_T$). The listing information is then encrypted using the key before it is sent by *L* to *T*. Although the identity data of *T* and *L* (ID_T, PK_T, P_T, ID_L, PK_L and P_L) are public, only they can generate the shared key $Key_{(L, T)}$ because they are the sole possessors of their secret keys (sk_T and sk_L). Then, *T* will use $Key_{(L, T)}$ to decrypt the listing information. Therefore, even if a hacker intercepts the information that was transmitted by *L* to *T*, they will be unable to decrypt the information.

4.5 Integrity

1. Definition: Integrity ensures that a file will not be changed, deleted, or damaged in any way during a transmission.
2. Scenario: A hacker attempts to change the contents of a listing or change the address of the digital currency address sent by *T* to *L* to gain illegal profit.
3. Solution: During step M2, *L* will publish information about the listing through the SC (Equation (34), $L \xrightarrow{Dapp} SC: modifierLL(createListing(ListingInfo))$). During this process, the *createListing*(*ListingInfo*) function is used to change the authorization information of the listing and store the hash of the listing's contents (*contentHash*) on the blockchain. Then, *T* will use *confirmOrder*() to validate the hash of the listing's contents during step T12 (Equation (60), $SC \rightarrow T: modifierT(confirmOrder(true), contentHash)$). The order will only be confirmed if the hashes match. After the order is confirmed, the SC will pay L through Step T14 (Equation (62), $SC \rightarrow L: payRent$).

4.6 Authenticity

1. Definition: Authenticity pertains to the ability to confirm the identity of a network user or information sender. In a public-key system, public keys can be used to verify the identities.
2. Scenario: A hacker intercepts the listing information sent by *L* to *T*, or impersonates *T*, stating that he/she has not received the listing information.
3. Solution: Both *T* and *L* must be registered with the CA, and the CA will place the registered participants on the blockchain network (Equations (9) and (10), R2. $CA \rightarrow T: verify(P_T, w_T)$ and R2′. $CA \rightarrow L: verify(P_L, w_L)$). *T* and *L* will also verify the identity of the other in step T7 (Equation (47), $T \leftrightarrow L: ID_T, PK_T, P_T \leftrightarrow ID_L, PK_L, P_L$), and then generate a shared key (Equation (50), $Key_{(L,T)} = sk_L \cdot PK_T$), which will be used to encrypt the listing information. Therefore, the hacker will be unable to impersonate *T* or decrypt the intercepted listing information.

4.7 Non-Repudiability

1. Definition: Non-repudiability is the impossibility of repudiating an event or behavior that has already occurred. In other words, any event that has occurred must carry a proof that prevents a repudiation.
2. Scenario: *L* acts like he/she has not received the digital currency paid by *T* and is asking *T* to pay a second time.
3. Solution: During the transaction between *T* and *L* (steps T1–T16), all information about the transaction will be openly stored on the blockchain. In step M2' (Equation (35), $T \xrightarrow{Dapp} SC: modifierTen(paymentNum, ID_T, PK_T, P_T)$), *T* will simultaneously generate the order and deposit their payment at the address of the SC. When *T* uses *confirmOrder*() to confirm the completeness of the order information, the SC will pay *L* in step T14 (Equation (62), $SC \rightarrow L: payRent$). Therefore, payment is performed immediately when the order was placed, and the transaction will be recorded on the blockchain, which provides non-repudiability.

4.8 Anonymity

1. Definition: Customer anonymity is the property that prevents the identity of a buyer or service user from being revealed during a transaction, or linked to a transaction (Pfitzmann & Köhntopp, 2001).
2. Scenario: When *T* sends an order to *L* through the SC, the SC must be able to validate the order without any knowledge about *T*. Furthermore, this lack of knowledge must not affect the transaction in any way.
3. Solution: Here, *T* will compute a ciphertext for order M and thus obtain C = $\{C_1, C_2\}$, and then generate a ciphertext digest (Equation (39), $h(C) = c$). The ciphertext digest will be blinded (Equation (40), $\Theta = c \cdot sk_T \cdot PK_T$) and thus the SC cannot read the contents of the transaction when signing the digest. Hence, *T* does not need to worry about data leakages from the signing process. Furthermore, when *L* receives the signed digest from the SC (Equation (53), $R - h(C) \times P_{SC} = e \cdot \mathrm{c} \cdot sk_T \cdot PK_T - c \cdot P_{SC}$), *L* will be able to decrypt the ciphertext (Equation (56), $M' = (c_{21} \cdot z_1^{-1}, c_{22} \cdot z_2^{-1}) = (y_1 \cdot m_1 \cdot z_1^{-1}, y_2 \cdot m_2 \cdot z_2^{-1}) = (m_1', m_2')$) and thus obtain the order in cleartext (Equation (58), $M' = M$).

4.9 Unforgeability

1. Definition: Unforgeability is a property that prevents tampering by a malicious third party during the transmission of data, and ensures that the data from the sender will reach the receiver without error.

2. Scenario: A hacker attempts to forge an order to fraudulently obtain a deposit payment.
3. Solution: *T* will encrypt the order, and then generate a ciphertext digest using a one-way hash function (Equation (39), $h(C) = c$). Because one-way hashes are irreversible, it is impossible for the hacker to obtain information about the order from the ciphertext digest. Furthermore, the forged ciphertext will fail the validation during the signature verification phase (Equation (53), $R - h(C) \cdot P_{SC} = e \cdot c \cdot sk_T \cdot PK_T - c \cdot P_{SC}$). Hence, it is impossible for a third party to forge an order.

4.10 Self-Certification Mechanism

1. Definition: The user will participate in the computation of the public key by the CA, and the certificate of the CA will be embedded within the public key, allowing other users to validate the public key of the user using this certificate.
2. Scenario: The hacker impersonates *L* or *T* using the CA to steal rent payments.
3. Solution: The participants will generate signature files using their identity information and random parameter (Equation (8), $V_{Ps} = h(d_{Ps} \parallel ID_{Ps})G$), and will only obtain a signature and public key after they have been registered with the CA (Equation (13), $w_{Ps} = k_{Ps} + sk_{CA}(q_{Psx} + h(ID_{Ps}))$. At this point, they will also be able to validate the public key. If all participants have obtained their signature and public key, they will then be able to verify the identity of the other using the public key provided by the CA, without maintaining a connection to the CA. During every phase, all of the participants are verifiable. All methods used by our protocol also satisfy the requirements of Girault's Level-3 security for public-key encryption systems (Girault, 1991).

In summary, our protocol minimizes third-party participation during all transactions, has a self-certification mechanism, and has security properties such as fairness, anonymity, resistance to double-spending and unforgeability. In terms of information security, our protocol satisfies all requirements for confidentiality, integrity, authenticity and non-repudiability. It also guarantees the privacy of its participants. The SC has also been designed to limit access to each function, while minimizing interruptions to transactions on the system and reducing the likelihood of an emergency. For instance, the conditional triggers of the SC have been designed in a way that prevents certain problems from occurring, which reduces the need for third-party conflict resolution mechanisms. Furthermore, the SC is fully transparent and openly accessible through the blockchain network, which is conducive to its use

by credible regulatory agencies. It should be noted that the aim of this paper is to highlight the feasibility of the proposed architecture; in the future, this architecture can be tested in a real system.

Please note that we have deliberately chosen not to compare our protocol with the SLA-based sharing economy service presented by Hang and Kim (2019) because the authors did not discuss their solution in terms of the aforementioned security metrics, nor did they describe the encryption mechanisms and algorithms they used. Furthermore, no mention was made regarding the security and privacy-enhancing functions of their method. In Table 2, we compare our protocol to other similar schemes in the literature, based on the ten aforementioned security metrics.

Table 2: Comparison between our scheme and other blockchain-based schemes based on ten security metrics

Security metric	Karamitsos et al. (2018)	Niya et al. (2018)	Li et al. (2018)	Liu et al. (2018)	Our scheme
Minimization of third-party participation	V	V	△	△	V
Fairness	V	V	╳	V	V
Privacy	—	—	V	V	V
Confidentiality	—	—	△	△	V
Integrity	V	V	△	△	V
Authenticity	—	—	△	△	V
Non-repudiability	V	V	V	V	V
Anonymity	—	—	V	V	V
Unforgeability	—	—	△	V	V
Self-certification mechanism	╳	╳	╳	╳	V
V: Fully satisfied; △: Partially satisfied; ╳: not satisfied; —: not applicable					

Although the SC-based protocols proposed by Karamitsos et al. (2018) and Niya et al. (2018) for house rentals minimize third-party participation and satisfy all requirements for fairness, integrity, and non-repudiability, the authors did not specify any algorithms that can handle transactions on their protocols. This makes it impossible to compare our protocol to their proposals in terms of transactional security. The FPPB method proposed by Li et al. (2018) uses zero-knowledge proofs

and stealth addresses to ensure transactional fairness in a sharing economy, which also guarantees transactional privacy without breaking the verification protocols or introducing off-blockchain interactions. Although the algorithms proposed for the FPPB ensure anonymity, they do not clearly show how third-party participation is minimized. The FPPB method also does not include an offline authentication mechanism. Liu et al. (2018) proposed a protocol that protects user privacy by breaking all links between users and property owners and ensures transactional fairness. This protocol uses algorithms that ensure fairness and anonymity, as well as a blind signature technique providing non-repudiability and unforgeability. However, it does not minimize third-party participation or provide offline authentication. The other protocols only consider fairness and anonymity, and overlook the minimization of third-party participation, confidentiality, integrity, and authenticity. Furthermore, none of the protocols apply an offline authentication or self-certification to strengthen the identity verification. Our protocol, by contrast, carries all of the advantages inherent to blockchain architectures, while being able to satisfy all of the aforementioned security requirements.

5. CONCLUSION AND SUGGESTIONS

The goal of this study was to develop a smart contract-based home-sharing scheme that uses a self-certification mechanism to perform identify verification in the matching and transaction phases, and to enhance security. Our transaction protocol minimizes third-party participation by using the automation provided by smart contracts and ensures transactional privacy through the use of a blind signcryption during the transaction phase for blinding all personal data. In addition to ensuring confidentiality, integrity, authenticity, and non-repudiability, all of the information that is transmitted by our protocol is encrypted, blinded, and hashed using a one-way hash function. This prevents data leakage or tampering, even if the data transmissions are intercepted by a third party. Hence, it is impossible for anyone other than the tenant or lessor to gain knowledge about the transactions conducted using our protocol. The proposed protocol also prevents third parties (such as financial institutes) from participating in a transaction, which significantly shortens the transaction process, and reduces the time costs of the protocol operation. Furthermore, this protocol possesses three of the most important features of a blockchain: the minimization of third-party participation, fairness, and privacy. It also provides confidentiality, integrity, authenticity, and non-repudiability, which are required by the information security standards of the ISO. The use of blockchain also minimizes the participation of unnecessary actors in the transaction, which enhances the trustworthiness of the sharing economy platform for its participants.

REFERENCES

王文宇 (2018)，「從共享經濟論 Airbnb 管制爭議」，*會計研究月刊*，第 394 期，頁 46-51。

沈淑惠 (2014)，「第三方支付研究」，*商學學報*，第 22 期，頁 21-40。

Airbnb. (2020). https://www.airbnb.com.

Asokan, N. (1998). Fairness in electronic commerce. Doctor of Philosophy, University of Waterloo, 24-35.

Botsman, R. & Rogers, R. (2010). What's mine is yours: The rise of collaborative consumption. Harper Business.

Chaum, D. (1984). Blind signature system. in Chaum, D. (Eds), *Advances in Cryptology*. Springer, Boston, MA, 153.

Chen, Y. L. (2018). Application of blockchain and smart contract for E-learning platform. Master's thesis, Tunghai University Computer Science Department.

Chiu, S. F. (2014). Disruption by the sharing economy: The construction of new service models. *Taiwan Economic Research Monthly*, 37(8), 18-24.

Constantiou, I., Marton, A., & Tuunainen, V. K. (2017). Four models of sharing economy platforms. *MIS Quarterly Executive*, 16(4), 231-251

Fan, M. & Zhang, X. (2019). Consortium blockchain based data aggregation and regulation mechanism for smart grid. *IEEE Access*, 7, 35929-35940.

Fromknecht, C., Velicanu, D., & Yakoubov, S. (2014). A decentralized public key Infrastructure with identity retention, https://allquantor.at/blockchainbib/pdf/fromknecht2014decentralized.pdf.

Girault, M. (1991). Self-certified public keys. in Davies, D.W. (Eds.), *Advances in Cryptology –EUROCRYPT 1991*, 547. Springer, Berlin, Heidelberg.

Hang, L. & Kim, D. H. (2019). SLA-based sharing economy service with smart contract for resource integrity in the Internet of Things. *Applied Sciences*, 9(17), 3602, 1-26.

ISO. (2005). Information technology-security techniques. Code of Practice for Information Security Management, ISO/IEC 17799.

Karamitsos, I., Maria, P., & Al Barghuthi, N. B. (2018). Design of the blockchain smart contract: A use case for real estate. *Journal of Information Security*, 9(3), 177-190.

Lai, C. H. (2003). The study of signcryption scheme based on elliptic curve. Master's thesis, Department of Computer Science and Information Engineering, Tamkang University.

Lee, J. Y. (2019). A decentralized token economy: How blockchain and cryptocurrency can revolutionize business. *Business Horizons*, 62(6), 773-784.

Lesavre, L., Varin, P., & Yaga, D. (2021). *Blockchain networks: Token design and management overview*. (No. NIST Internal or Interagency Report (NISTIR) 8301). National Institute of Standards and Technology.

Li, B., Wang, Y., Shi, P., Chen, H., & Cheng, L. (2018). FPPB: A fast and privacy-preserving method based on the permissioned blockchain for fair transactions in sharing economy. *IEEE International Conference on Trust*, 1368-1373.

Liu, Z., Li, Y., Liu, Y., & Yuan, D. (2018). Sharing economy protocol with privacy preservation and fairness based on blockchain I. *In International Conference on Cloud Computing and Security*, Springer, 59-69.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Niya, S. R., Schüpfer, F., Bocek, T., & Stiller, B. (2018). A Peer-to-peer Purchase and Rental Smart Contract-based Application (PuRSCA). *it-Information Technology*, 60(5-6), 307-320.

OWASP. (2017). Category: OWASP top ten project, https://owasp.org/www-project-top-ten/.

Pfitzmann, A. & Köhntopp, M. (2001). Anonymity, unobservability, and pseudonymity–A proposal for terminology. *In Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, 1-9.

Risius, M. & Spohrer, K. (2017). A blockchain research framework. *Business and Information Systems Engineering*, 59(6), 385-409.

Szabo, N. (1994). Smart contracts, http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

Tung W. F. & Chiu Y. F. (2019). Transnational sentiment analysis of social media for CSA social enterprise innovation - from the perspective of sharing economy and collaborative consumption. *Journal of Information Management*, 26(1), 71-98

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). Blockchain technology overview. *arXiv preprint arXiv*:1906.11078.

Yu, T.Y. (2020). A novel electronic payment scheme based on smart contracts for digital content transactions. Master's thesis, Department of Information Research, National Defense University.

Yuan, Y., & Wang, F. Y. (2016). Blockchain: The state of the art and future trends. *Acta Automatica Sinica*, 42(4), 481-494.

Zheng, Y. (1997). Digital signcryption or how to achieve cost. *Advances in Cryptology-Crypto*, 97, 1294, 165-179.