Abstract: In recent years, due to the slowdown of the global economy and the instability of the stock market and real estate market, the art market has gradually become the third-largest investment market after the former two. As more and more funds flow into the art market, the two seemingly unrelated industries, art, and finance have increasingly close cooperation due to the growing prosperity of the art market. As a very important part of the process of art financialization, the development of art banks has also received extensive attention from all walks of life. In addition, blockchain, as a technology to jointly maintain reliable databases through trustlessness and decentralization, has grown rapidly in recent years, and is gradually applied in various fields. The advantages of blockchain technology are decentralization, anonymity, immutability, and traceability of stored information. The proposed scheme applies the characteristics of blockchain technology to Art Bank’s art rental system, and will have the following advantages: the use of Hyperledger technology, various art banks being merged into an alliance, and information among alliance members being shared, which is convenient for tenants to comprehensively query a single item. Using blockchain technology, in an environment where there is no central authority, under the premise of preventing the tampering of artwork-related information, ensures that the detailed information of the artwork is shared. Traditional leasing agreements can be compiled into smart contract leasing contracts to automatically run and manipulate data. The proposed protocol satisfies the following security requirements: identities’ mutual authentication, non-repudiation between every two parties, and other major security requirements based on blockchain. When a dispute arises, our proposed scheme also has an arbitration mechanism to clarify responsibilities.

Keywords: smart contract; blockchain; mutual authentication; art bank; Hyperledger Fabric; arbitration mechanism; non-repudiation

1. Introduction

1.1. Background

An art bank refers to a professional bank that invests in artworks. It is a financial investment institution with art investment as its core asset. The main business of Art Bank is to realize investment income through the use of the art trading market. From the origin of art banks, it is inevitable.

First, there is huge financial support. Driven by financial liberalization, internationalization, and financial innovation in the international market, a large amount of hot money has emerged. Every day, nearly USD 1000 billion in hot money is seeking a home in the world market. If the government uses part of the national savings to invest in art, it is also a huge sum of money, which can provide huge financial support to the art bank [1,2].

Second, there is a wide range of social needs. To date, many businesses and entrepreneurs have increasingly favored the use of art investment as a business strategy due to art investment with a very strong advertising effect. In the eyes of the public, the
palace of art is sacred, and elegant investments can leave a deep impression on the hearts of consumers.

In addition, art investment is conducive to the construction of spiritual civilization and the development of social welfare undertakings. Thus, it is easily spread by the news media, and its publicity effect is not only an advertisement, but is better than an advertisement. It can be seen that the use of art investment as a business strategy can not only establish a good cultural image for enterprises and show strong economic strength to society but also promote the popularization and development of art. It will also bring good economic benefits to the enterprise through good social benefits [3,4].

Finally, an art collection can maintain and increase its value. The world-famous oil painting “Mona Lisa” was bought by King Francis of France in 1517 for 492 taels of gold (equivalent to USD 270,000) and is now worth hundreds of millions of dollars. In the international art market, the works of artists recognized by connoisseurs generally increase in value by 30% to 50%, and some even as high as several hundred percent [5].

Therefore, it is often referred to as “soft gold” around the world. With the development of the social economy, the demand for artworks will continue to increase, and artworks are generally scarce and irreplaceable, and price increases become inevitable.

Another important function of the art bank is to absorb and make profits through the bank’s operating model. First of all, the art bank formed by shareholder investors’ equity participation and investment, absorbs the works of many art collectors in the society to the art bank and then uses the methods of selling and leasing artworks to make profits. The absorption of artworks from individuals or legal entities in society can be achieved through various channels, such as purchasing artworks with value-added potential from individuals and legal entities in society to achieve the purpose of increasing capital. For individuals or legal entities who are willing to deposit their artworks in art banks, a certain interest rate is granted to art collectors to maintain the number of collections, so that they can lease artworks to enterprises and government departments for display and collect rent [6,7].

Through the explanation of the above content, we can realize that the current generation is an information age, where information has a very high value. The current world is a society where the big ones are always big. When a company has a leading position in a certain field, it may become the dominant company in that field. Before discussing art rentals, let us try to discuss housing rentals. When a housing rental company has a large enough inventory, it may become a seller-led state. Leasing companies are free to change rental prices and control or even monopolize the housing rental market [8–10].

In the existing housing rental environment, the Internet usually only provides a basic introduction to housing and cannot fully record the rental process. If the lessee is interested in a certain house item, he/she will contact the landlord and sign a lease contract with the landlord to protect each other’s rights and interests. When the lessee pays the rent to the landlord, it is also possible to use a third-party payment tool, which cannot achieve an irreversible transaction process and cannot completely preserve the transaction records generated during the lease period. Therefore, in the existing housing rental environment, it is inevitable that there will be lease disputes between the landlord and the tenant. These conditions that exist in the housing rental environment also exist in the rental environment of high-priced artworks [11–14].

In recent years, the emerging technology of blockchain has been shown to resolve the shortcomings of the current leasing environment (whether it is art or housing). Being centralized and controlled by large leasing companies means that the leasing process cannot be open and transparent. Blockchain uses peer-to-peer technology, which can provide the advantage of decentralization. The workload proof of blockchain technology can ensure that the information stored in each block will not be easily tampered with. In addition, blockchain technology also extends the application of smart contracts. A smart contract is a program script suitable for the decentralized architecture of the blockchain. When a specific condition in the smart contract is triggered, subsequent related programs will automatically be executed to complete the overall script [15–18]. For research on
the application of blockchain technology in the business environment, Chen et al. also proposed related research [19] based on blockchain for the anti-switch package issue of commercial transactions.

The proposed scheme applies the characteristics of blockchain technology to Art Bank’s artwork rental system, and obtains the following advantages [20–24]:

a. Using the Hyperledger technology, the various art banks are combined into a single alliance, and information among alliance members can be shared, which facilitates the integrated query of a single entry by the renter.

b. Use blockchain technology to ensure that the details of artwork information are shared under the premise of preventing the tampering of artwork-related information in an environment without a central authority.

c. Compile traditional lease agreements into smart contract lease contracts to automatically run and operate data.

d. When a lease dispute occurs, a third-party arbitration institution can conduct a fair arbitration judgment based on the information stored in the blockchain.

1.2. Problem Formulation

This research will apply blockchain technology, combined with smart contracts for automatic script control, and propose an art rental system. Like lots of blockchain-related studies proposed by researchers in the past [25–35], the architecture and communication protocols proposed in this study must meet the following goals:

(1) Decentralization and information sharing: Under the decentralized environment framework of the blockchain, all nodes in the environment have the same rights and obligations, and can store and share information. Therefore, data errors or failure shutdowns of a few nodes in the environment will not affect the operation of the entire blockchain network.

(2) Openness: The blockchain has a public key system. We can use the private key to encrypt private information, and allow others to query and verify the correctness of the open blockchain data through the public key. In the blockchain environment, blockchain data will be open and transparent, reducing public distrust caused by information asymmetry. Through the open data of the blockchain architecture, the related applications of big data and cloud computing can be improved.

(3) Privacy and access control: Even with the application of blockchain technology, extremely high levels of privacy can be achieved. Blockchain technology can also combine public key encryption and access control. Only allowed members can access and verify data. This is the concept of a consortium chain or private chain. Different from the public chain, through the alliance chain or private chain, only members in the chain can access the data and combine it with smart contracts for automatic script control.

(4) Mutual authentication: Mutual authentication is a basic security verification condition in the field of information security. This is used to confirm whether a certain framework or communication protocol can verify the legitimacy of the identities of the communication parties. In the related research of researchers, Burrows–Abadi–Needham (BAN) logic proof is often used to judge whether the architecture or communication protocol has realized the mutual authentication of the communication parties.

(5) Verifiable: Verifiability refers to a certain architecture or communication protocol. During the communication process, the message sent by the sender will be signed by the sender’s private key. After the message is sent to the receiver, the receiver can use the sender’s public key to verify that the sender’s signature is authentic.

(6) Integrity and forgery: In related research, to guarantee the integrity of a certain framework or communication protocol during information transmission, a public key system is usually used to ensure that the transmitted information has not been tampered with by malicious persons.
(7) Traceable: The use of blockchain technology means that information transmission is equal, and all participating nodes in the system have equal rights and obligations, so all nodes can cross-check transaction information on the system. In addition, due to the decentralized nature of blockchain technology, the change records of a certain node on the system will be saved by other nodes, achieving traceability.

(8) Non-repudiation: The blockchain system has undeniable characteristics. After the information transmitted in the system is verified by a certain role and transmitted to the blockchain, the information will be permanently stored. In addition to verifying the correctness of the original message, this information also records the recording time of the message. If we want to change a message on the blockchain, we must change the content of more than half of the nodes in the system at the same time, which will be difficult to achieve under the open system architecture.

(9) Resist man-in-the-middle attack: A man-in-the-middle attack means that an attacker can intercept a message during sending, and modify or resend an illegal message to the receiver. Under the framework of the blockchain, the message sent by the sender will be signed with its private key, and the attacker cannot obtain the private key of the message sender, so it will be difficult for the attacker to implement a man-in-the-middle attack.

After the background introduction of this study, we organize the rest of the content as follows. The relevant preliminary knowledge to facilitate the understanding of this study will be introduced in Section 2. The detailed architecture and communication protocol proposed in this research will be introduced in Section 3. For the communication protocol proposed in this research, we analyzed the characteristics and security of the blockchain, which will be introduced in Section 4. For the communication protocol proposed in this study, the analysis of the related computational cost and the related communication cost, and the comparison with the existing blockchain research related to leasing, will be introduced in Section 5. Last, the conclusion of this research and future work will be presented in Section 6.

2. Preliminary

2.1. Smart Contract

In 1996, Nick Szabo [36,37] proposed a smart contract, which is a program script for realizing intelligent automatic control in blockchain technology. Smart contracts are controlled by program logic, triggering program codes through specific conditions in the system, and automatically executing the complete program content under the conditions. Through smart contracts, both roles in a transaction can complete a traceable and irreversible transaction without the intervention of a third party. Smart contracts can also support an automatic verification mechanism. When both parties want to verify the transaction, they can also trigger the smart contract to execute the program content of specific conditions to implement a transparent and reliable transaction verification mechanism.

2.2. ECDSA

Vanstone [38,39] proposed the Elliptic Curve Digital Signature Algorithm (ECDSA) in 1992. It is a derivative of the Digital Signature Algorithm (DSA) using Elliptic Curve Cryptography (ECC). The characteristics of ECC make ECDSA require a significantly smaller key size and the same security level, thereby providing faster calculations and less storage space. The security of the elliptic curve digital signature algorithm (ECDSA) is based on the discrete logarithm problem.

Next, we briefly describe the three phases of ECDSA key generation for verification:

Key generation phase: We assume that any participant must apply to our blockchain center for public and private keys, the key generation with ECDSA is as follows: \( Q_x = d_xG \), where \( x \) is the participant ID, \( Q_x \) is the public key, \( d_x \) is the private key, and \( G \) is a generating point based on the elliptic curve. The public key \( Q_x \) and private key \( d_x \) are sent to the participant and stored. \( Q_x \) will also be stored in the blockchain center.
Signature phase: Assume that there is a message needed to be sent by participant x to y. x chooses a random number k between 1 and n-1, and calculates a point on the curve as follows \((x, y) = k \times G\). Then \(r_x = x \mod n\) is calculated. Next, x signs message m as follows:
\[
sig_x = k - 1 (h(m) + r_x d_x)
\]
sending \((m, r_x, \sig_x)\) to y. Verification phase: When y receives \((m, r_x, \sig_x)\), calculated parameters are as follows:
\[
(x', y') = (h(m) \sig_x^{-1} \mod n)G + (r_x \sig_x^{-1} \mod n)Q_x.
\]
Then, \(x' \equiv r_x \mod n\) is verified to determine if the signature is valid.

2.3. BAN Logic

Burrows-Abadi-Needham (BAN) logic [40,41] was proposed by Burrows et al. in 1990. BAN logic is belief logic, and the purpose of BAN logic is to analyze authentication protocols by deriving beliefs. Therefore, it is suitable to define and analyze whether the communication parties have reached mutual authentication during the communication process.

2.4. Hyperledger Fabric

Blockchains are mainly divided into public blockchains, private blockchains, and consortium blockchains. The data of the public chain is completely open, and all participating nodes can obtain the data, which is not suitable for private transactions; the private chain is too centralized and only suitable for the internal data management of a single organization; and the alliance chain is the most flexible one, and as long as the participating nodes can form a consortium, the data will only circulate in this alliance, and Hyperledger Fabric is a kind of alliance chain blockchain. Hyperledger Fabric is an open-source system. This project was created and managed by the Linux Foundation, which is committed to creating a transparent, secure, and decentralized enterprise-level blockchain solution [42–44]. Compared with public chains and private chains, Hyperledger Fabric has lower computing costs, high efficiency, architectural flexibility, scalability, and confidentiality.

3. The Proposed Scheme

3.1. System Architecture

This research proposed an artwork rental system based on blockchain technology. The main roles of the framework include Certificate Authority, Blockchain Center, Art Bank, Insurance Company, Bank, and Licensee. The system architecture is shown in Figure 1.

![System architecture diagram](image-url)
Step 1: Certificate Authority (CA), Art Bank (AB), Insurance Company (IC), Bank (BK), and Licensee (LE) must first register with the Blockchain Center to obtain the public and private keys for information encryption and decryption, and also Elliptic Curve Digital Signature Algorithm (ECDSA) signature key.

Step 2: A large number of Art Banks form a consortium for resource exchange and sharing. The licensee finds the collection that he/she wants to rent through an integrated query, and then contacts the Art Bank. The licensee informs of the artwork to be rented, as well as the time and place of display, and Art Bank replies with the value certificate and rent of the artwork, as well as the payment account to the licensee. All relevant information is uploaded to the Blockchain Center through Certificate Authority.

Step 3: The licensee informs the Insurance Company that he/she wants to rent artwork from Art Bank and provides the value certificate of the artwork. After the Insurance Company confirms, it informs the insurance amount that can be provided, insurance cost, and payment account number. All relevant information is uploaded to the Blockchain Center through Certificate Authority.

Step 4: The licensee pays through the bank, which includes the rent paid to Art Bank and the insurance fee paid to the Insurance Company. The bank will send the relevant payment receipt back to the licensee. All relevant information is uploaded to the Blockchain Center through Certificate Authority.

Step 5: The licensee informs Art Bank that the rent and insurance fees have been paid, and provides relevant payment receipts. Art Bank verifies whether the payment receipt provided by the licensee is consistent with the information in the Blockchain Center, and if confirmed, will rent the artwork in accordance with the licensee’s requirements. All relevant information is uploaded to the Blockchain Center through Certificate Authority.

3.2. Notation

The following is the notation of this research:

- \( q \): A k-bit prime number
- \( GF(q) \): A finite group \( q \)
- \( E \): An elliptic curve that is defined on the finite group \( q \)
- \( G \): A generating point that is based on the elliptic curve \( E \)
- \( PK_x \): An asymmetric public key of role \( x \)
- \( SK_x \): An asymmetric private key of role \( x \)
- \( ID_x \): A name that represents identity \( x \)
- \( k_{x,y} \): A random value which on the elliptic curve of \( x \) to \( y \)
- \((r_{xy}, s_{xy})\): An elliptic curve signature value of \( x \) to \( y \)
- \( M_{xy} \): A message that is transmitted from \( x \) to \( y \)
- \( ID_{BC_x} \): An index value of the blockchain information
- \( BC_{xy} \): Blockchain information of \( x \) to \( y \)
- \( TS_{xy} \): A timestamp message of \( x \) to \( y \)
- \( \Delta T \): Checks that if the difference between two timestamps is less than a specified timestamp
- \( E_{PK_x}(M) \): Encrypt an information \( M \) using the asymmetric public key of \( x \)
- \( D_{SK_x}(M) \): Decrypt an information \( M \) using the asymmetric private key of \( x \)
- \( Enc_{x,y} \): Encrypted information that uses the asymmetric key of \( x \) to \( y \)
- \( Cert_x \): A certificate of role \( x \) that conforms to the X.509 standard
- \( BK_x \): A bank account of role \( x \)
- \( RT_x \): A payment receipt of role \( x \)
- \( AB_x \): An artwork rent contract of role \( x \)
- \( AB_{TID} \): A transaction number that is issued by an art bank
- \( IC_x \): An artwork insurance contract of role \( x \)
- \( IC_{TID} \): A transaction number which is issued by an insurance company
- \( h(.) \): A hash function
- \( A \equiv B \): A judgmental that verifies whether \( A \) is equal to \( B \)
3.3. Registration Phase

The system role X can represent the Certificate Authority (CA), Licensee (LE), Art Bank (AB), Insurance Company (IC), and Bank (BK). These parties are registered with the Blockchain Center (BCC), which then obtain a digital certificate of identity and a relative public/private key pair from the BCC via a secure channel. The flowchart of the registration phase is shown in the following Figure 2.

![Figure 2. Registration phase.](image)

- **Step 1**: An identity \( ID_X \) is generated by role X, and is sent to the blockchain center.
- **Step 2**: An ECDSA private key \( d_X \), which is based on role X, is generated by the blockchain center, and \( Q_X = d_X G \) is calculated. Then, \( ID_X, Cert_X, PK_X, SK_X, (d_X, Q_X) \) will be transmitted to role X by the blockchain center.
- **Step 3**: Finally, \( (Cert_X, PK_X, SK_X, d_X, Q_X) \) is stored by role X.

3.4. Smart Contract Initialization

In our proposed architecture, we apply blockchain technology. During the authentication and authorization process, some key information is saved and verified through the BCC. Also, in the smart contract, we define the key information in the blockchain. The blockchain smart contract structure for the proposed scheme is shown in the following Figure 3.

In our proposed smart contract, we develop some key information that is stored in the blockchain. In each smart contract, we present the basic fields of ID (identification), certificate, transaction detail, and timestamp. The artwork rental contract is added in the AB_LEinf/LE_BKinf/LE_ABinf smart contract, while the artwork insurance contract is added in the IC_LEinf/LE_BKinf/LE_ABinf smart contract. At last, the payment receipt is shown in the BK_LEinf/LE_ABinf smart contract. The public and private key pairs for all roles are also issued by the blockchain center in the registration phase.

3.5. ECDSA Authentication Process

In our proposed scheme, before the beginning of the communication, system role A and role B must first verify the legitimacy of each other’s identities through the ECDSA mechanism. System role A or role B can represent the Certificate Authority (CA), Licensee (LE), Art Bank (AB), Insurance Company (IC), and Bank (BK). The flowchart of the ECDSA authentication process is shown in the following Figure 4. The ECDSA signature process is shown in the following Algorithm 1, and the ECDSA verification process is shown in the following Algorithm 2.

- **Step 1**: A random value \( k_{α−β} \) is generated by role A,

\[
\text{token} = h(ID_α, M_{α−β}, TS_{α−β})
\]
is calculated, \((token, k_{α−β}, d_α)\) is passed to the signature process of Algorithm 1, and \((r_{α−β}, s_{α−β})\) is obtained.

\[
Enc_{α−β} = E_{PK_β}(ID_α, M_{α−β}, TS_{α−β})
\]  

(3)
is calculated, and \(ID_α, Enc_{α−β}, (r_{α−β}, s_{α−β})\) is sent to role B.

**Algorithm 1.** Role A and role B’s ECDSA signature process

```
str token, k_{α−β}, d_α;
func signature process (token, k_{α−β}, d_α)
    z_{α−β} = token;
    (x_{α−β}, y_{α−β}) = k_{α−β}G;
    r_{α−β} = x_{α−β}modn;
    s_{α−β} = k_{α−β}^{-1}(z_{α−β} + r_{α−β}d_α)modn;
    rtn (r_{α−β}, s_{α−β});
```

Figure 3. Smart contract structure of the proposed scheme.
Figure 4. ECDSA authentication process.

Step 2:

\[(ID_A, M_{A-B}, T_{SA-B}) = D_{SK_B}(Enc_{A-B}) \quad (4)\]

is first calculated by role B, \(T_{SNOW} - T_{S_{A-B}} \leq \Delta T\) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified.

\[\text{token} = h(ID_A, M_{A-B}, T_{SA-B}) \quad (5)\]

is calculated, the verification process by \((\text{token}, r_{A-B}, s_{A-B}, Q_a)\) as Algorithm 2 is called, and a result of valid/invalid is obtained.

**Algorithm 2.** Role A and role B’s ECDSA verification process

```
str token, r_{A-B}, s_{A-B}, Q_a;
func verification process (token, r_{A-B}, s_{A-B}, Q_a)
    z_{A-B} = token;
    u_{A-B1} = z_{A-B} s_{A-B}^{-1} \mod n;
    u_{A-B2} = r_{A-B} s_{A-B}^{-1} \mod n;
    (x_{A-B}', y_{A-B}) = u_{A-B1} G + u_{A-B2} Q_a;
    chk if x_{A-B}' = r_{A-B} \mod n;
    rtn Valid/Invalid;
```
A random value \( k_{\beta - a} \) is generated by role B, and

\[
\text{token} = h(ID_{\beta}, M_{\beta - a}, TS_{\beta - a})
\]  

(6)

is calculated, \( (\text{token}, k_{\beta - a}, d_{\beta}) \) is sent to the signature process, and \( (r_{\beta - a}, s_{\beta - a}) \) is obtained.

\[
Enc_{\beta - a} = E_{PK_{\beta}}(ID_{\beta}, M_{\beta - a}, TS_{\beta - a})
\]  

(7)

is calculated, and \( ID_{\beta}, Enc_{\beta - a}, (r_{\beta - a}, s_{\beta - a}) \) sent to role A.

Step 3:

\[
(ID_{\beta}, M_{\beta - a}, TS_{\beta - a}) = D_{SK_{\beta}}(Enc_{\beta - a})
\]  

(8)

is first calculated by role A, \( TS_{NOW} - TS_{\beta - a} \leq \Delta T \) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified.

\[
\text{token} = h(ID_{\beta}, M_{\beta - a}, TS_{\beta - a})
\]  

(9)

is calculated, the \( (\text{token}, r_{\beta - a}, s_{\beta - a}, Q_{\beta}) \) is passed to the verification process, and a result of valid/invalid is obtained.

3.6. CA Communication Process

The Hyperledger blockchain architecture is applied in our proposed method, increasing the role of CA, which allows more flexible access control, and also reduces the burden of BCC. The data will be sent to their respective CA after verifying communication between each role, and then the blockchain data will be transmitted to the BCC through each CA. For example, all Art Banks have their own CA, which can ensure the data sharing of each CA member, as well as cross-CA access control, taking into account privacy and efficiency. A different member of Art Banks can be represented by the Access Party (AP). The flowchart of the CA communication process is shown in the following Figure 5.

Step 1: A random value \( k_{AP - CA} \) is generated by AP,

\[
\text{token} = h(ID_{AP}, M_{AP - CA}, Cert_{AP}, TS_{AP - CA}, ID_{BC})
\]  

(10)

is calculated, the ECDSA signature process by \( (\text{token}, k_{AP - CA}, d_{AP}) \) is called, and \( (r_{AP - CA}, s_{AP - CA}) \) is obtained.

\[
Enc_{AP - CA} = E_{PK_{CA}}(ID_{AP}, M_{AP - CA}, Cert_{AP}, TS_{AP - CA}, ID_{BC})
\]  

(11)

is calculated, and \( ID_{AP}, Enc_{AP - CA}, (r_{AP - CA}, s_{AP - CA}) \) sent to CA.

Step 2:

\[
(ID_{AP}, M_{AP - CA}, Cert_{AP}, TS_{AP - CA}, ID_{BC}) = D_{SK_{CA}}(Enc_{AP - CA})
\]  

(12)

is first calculated by CA, \( TS_{NOW} - TS_{AP - CA} \leq \Delta T \) is used to confirm whether the timestamp is valid, \( Cert_{AP} \) is verified with \( PK_{AP} \), and then the correctness of the ECDSA signature is verified.

\[
\text{token} = h(ID_{AP}, M_{AP - CA}, Cert_{AP}, TS_{AP - CA}, ID_{BC})
\]  

(13)

is calculated, the verification process by \( (\text{token}, r_{AP - CA}, s_{AP - CA}, Q_{AP}) \) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from AP will be received by CA and the smart contracts AP_CAins and AP_CAdck will be triggered. Algorithm 3 is presented as follows:
Algorithm 3. The proposed scheme’s smart contract AP_CAins and AP_CAchk

```plaintext
func ins SC AP_CAins (str APid, str APdetail, str APcert, str APtsp)
    cnt ++;
    AP[cnt].id = id;
    AP[cnt].detail = detail;
    AP[cnt].cert = cert;
    AP[cnt].tsp = tsp;
}
sign str APkey (APid, APdetail, APcert, APtsp);
verify str APkey (APid, APdetail, APcert, APtsp);
func chk SC AP_CAchk (str APid, str APdetail, str APcert, str APtsp)
    rtn APid.exist;
    rtn APdetail.exist;
    rtn APcert.exist;
    rtn APtsp.exist;
```

\[
BC_{AP-CA} = h(r_{AP-CA}, s_{AP-CA})
\] (14)

is calculated by CA, \((ID_{BC}, BC_{AP-CA})\) and will also be uploaded to the blockchain center. Then, a random value \(k_{CA-AP}\) is generated by CA, and

\[
token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, T_{CA-AP}, ID_{BC})
\] (15)

is calculated, the ECDSA signature process by \((token, k_{CA-AP}, d_{CA})\) is called and \((r_{CA-AP}, s_{CA-AP})\) is obtained.

\[
Enc_{CA-AP} = E_{PK_{AP}}(ID_{CA}, M_{CA-AP}, Cert_{CA}, T_{CA-AP}, ID_{BC})
\] (16)

is calculated, and \(ID_{CA}, Enc_{CA-AP}, (r_{CA-AP}, s_{CA-AP})\) is sent to AP.

Step 3:

\[
(ID_{CA}, M_{CA-AP}, Cert_{CA}, T_{CA-AP}, ID_{BC}) = D_{SK_{AP}}(Enc_{CA-AP})
\] (17)

is first calculated by AP, \(T_{NOW} - T_{CA-AP} \leq \Delta T\) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified.

\[
token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, T_{CA-AP}, ID_{BC})
\] (18)

is calculated, the verification process by \((token, r_{CA-AP}, s_{CA-AP}, Q_{CA})\) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from AP will be received by CA and the smart contracts CA_APins and CA_APchk will be triggered. Algorithm 4 is presented as follows:

Algorithm 4. The proposed scheme’s smart contract CA_APins and CA_APchk

```plaintext
func ins SC CA_APins (str CAid, str CAdetail, str CAcert, str CATsp)
    cnt ++;
    CA[cnt].id = id;
    CA[cnt].detail = detail;
    CA[cnt].cert = cert;
    CA[cnt].tsp = tsp;
}
sign str CAkey (CAid, CAdetail, CAcert, CATsp);
verify str CAkey (CAid, CAdetail, CAcert, CATsp);
func chk SC CA_APchk (str CAid, str CAdetail, str CAcert, str CATsp)
    rtn CAid.exist;
    rtn CAdetail.exist;
    rtn CAcert.exist;
    rtn CATsp.exist;
```
is calculated by AP, and \((ID_{BC}, BC_{CA-AP})\) will also be uploaded to the blockchain center.

\[ BC_{CA-AP} = h(r_{CA-AP}, s_{CA-AP}) \]  

3.7. Artwork Rental Phase

The licensee (LE) finds the collection that he/she wants to rent through an integrated query, and then contacts the Art Bank (AB). The LE informs AB which artwork is to be rented, as well as the time and place of display. The AB will first verify the identity of the LE and replies with the value certificate and rent of the artwork, as well as the payment account to the LE. The LE should then apply for insurance from the insurance company, and pay the artwork rent and insurance costs through the bank. We present the flowchart of the artwork rental phase in Figure 6.
A random number \( k_{LE,AB} \) is chosen and
\[
\text{token} = h(ID_{LE}, M_{LE,AB}, Cert_{LE}, TS_{LE,AB}, ID_{BC})
\]
The signature process \((\text{token}, s_{LE,AB}, d_{LE})\) is called, and \( (r_{LE,AB}, s_{LE,AB}) \) is obtained,
\[
\text{Enc}_{LE,AB} = E_{PK_{AB}}(ID_{LE}, M_{LE,AB}, Cert_{LE}, TS_{LE,AB}, ID_{BC})
\]
Figure 6. Artwork rental phase.

Step 2: \( (ID_{AB}, M_{AB,LE}, Cert_{AB}, AB_{ID}, AB_{LE}, TS_{AB,LE}, ID_{BC}) = D_{SK_{AB}}(\text{Enc}_{AB,LE}) \)

is calculated, and \(\text{Enc}_{AB,LE} \) is sent to AB.

\[
(ID_{LE}, M_{LE,AB}, Cert_{LE}, TS_{LE,AB}, ID_{BC}) = D_{SK_{AB}}(\text{Enc}_{LE,AB})
\]
is first calculated by AB, and then the correctness of the ECDSA signature is verified.
is calculated, the verification process by \((token, r_{LE-AB}, s_{LE-AB}, Q_{LE})\) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from LE will be received by AB, and the smart contracts LE_ABins and LE_ABchk will be triggered. Algorithm 5 is presented as follows:

**Algorithm 5.** The proposed scheme’s smart contract LE_ABins and LE_ABchk

```java
func ins SC_LE_ABins (str LEid, str LEdetail, str LEcert, str LETsp) {
    cnt ++;
    LE[cnt].id = id;
    LE[cnt].detail = detail;
    LE[cnt].cert = cert;
    LE[cnt].tsp = tsp;
}
sign str LEkey (LEid, LEdetail, LEcert, LETsp);
verify str LEkey (LEid, LEdetail, LEcert, LETsp);
func chk SC_LE_ABchk (str LEid, str LEdetail, str LEcert, str LETsp) {
    rtn LEid.exist;
    rtn LEdetail.exist;
    rtn LEcert.exist;
    rtn LETsp.exist;
}
```

is calculated by AB, \((ID_{BC}, BC_{LE-AB})\) and will also be uploaded to the blockchain center. Then, a random value \(k_{AB-LE}\) is generated by AB, and

\[token = h(ID_{LE}, M_{LE-AB}, Cert_{LE}, TS_{LE-AB}, ID_{BC})\]  \hspace{1cm} (23)

is calculated, the verification process by \((token, r_{LE-AB}, s_{LE-AB}, Q_{LE})\) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from LE will be received by AB, and the smart contracts LE_ABins and LE_ABchk will be triggered. Algorithm 6 is presented as follows:

**Algorithm 6.** The proposed scheme’s smart contract AB_LEins and AB_LEchk

```java
BC_{LE-AB} = (r_{LE-AB}, s_{LE-AB})\]  \hspace{1cm} (24)
```

is calculated by LE, \((ID_{AB}, M_{AB-LE}, Cert_{AB}, ABTID, ABLE, TS_{AB-LE}, ID_{BC})\) is calculated, the signature process by \((token, k_{AB-LE}, d_{AB})\) is called, and \((r_{AB-LE}, s_{AB-LE})\) is obtained.

\[Enc_{AB-LE} = E_{PK_{LE}}(ID_{AB}, M_{AB-LE}, Cert_{AB}, ABTID, ABLE, TS_{AB-LE}, ID_{BC})\]  \hspace{1cm} (25)

is calculated, the CA communication process is called, and \(ID_{AB}, Enc_{AB-LE}, (r_{AB-LE}, s_{AB-LE})\) is sent to LE.

**Step 3:**

\((ID_{AB}, M_{AB-LE}, Cert_{AB}, ABTID, ABLE, TS_{AB-LE}, ID_{BC}) = D_{SK_{LE}}(Enc_{AB-LE})\]  \hspace{1cm} (26)

is first calculated by LE, \(T_{NOW} - TS_{AB-LE} \leq \Delta T\) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified.

\[token = h(ID_{AB}, M_{AB-LE}, Cert_{AB}, ABTID, ABLE, TS_{AB-LE}, ID_{BC})\]  \hspace{1cm} (27)

is calculated, the verification process by \((token, r_{AB-LE}, s_{AB-LE}, Q_{LE})\) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from AB will be received by LE and the smart contracts AB_LEins and AB_LEchk will be triggered. Algorithm 6 is presented as follows:

\[BC_{AB-LE} = (r_{AB-LE}, s_{AB-LE})\]  \hspace{1cm} (28)

is calculated by LE, and \((ID_{BC}, BC_{AB-LE})\) will also be uploaded to the blockchain center.
Algorithm 6. The proposed scheme's smart contract AB_LEins and AB_LEchk

```plaintext
func ins SC AB_LEins (str ABid, str ABdetail, str ABcert, str AB_LEABtid, str ABtsp) {
    cnt ++;
    AB[cnt].id = id;
    AB[cnt].detail = detail;
    AB[cnt].cert = cert;
    AB[cnt].LEABtid = LEABtid;
    AB[cnt].tsp = tsp;
}

sign str ABkey (ABid, ABdetail, ABcert, AB_LEABtid, ABtsp);
verify str ABkey (ABid, ABdetail, ABcert, AB_LEABtid, ABtsp);

func chk SC AB_LEchk (str ABid, str ABdetail, str ABcert, str AB_LEABtid, str ABtsp) {
    rtn ABid.exist;
    rtn ABdetail.exist;
    rtn ABcert.exist;
    rtn AB_LEABtid.exist;
    rtn ABtsp.exist;
}
```

3.8. Insurance Insured Phase

After the licensee (LE) and art bank (AB) negotiate an artwork rental contract, the LE must ensure the artwork from the insurance company (IC). The IC will confirm the rent contract between the LE and AB, and inform the LE of the insurance content and the insurance fees that need to be paid. We present the flowchart of the insurance insured phase in Figure 7.

Step 1: A random value $k_{LE-IC}$ is generated by the LE,

$$token = h(ID_{LE}, M_{LE-IC}, Cert_{LE}, AB_{LE}, TS_{LE-IC}, ID_{BC})$$ (30)

is calculated, the signature process by $(token, k_{LE-IC}, d_{LE})$ is called, and $(r_{LE-IC}, s_{LE-IC})$ is obtained.

$$Enc_{LE-IC} = E_{PK_{IC}}(ID_{LE}, M_{LE-IC}, Cert_{LE}, AB_{LE}, TS_{LE-IC}, ID_{BC})$$ (31)

is calculated, and $ID_{LE}, Enc_{LE-IC}, (r_{LE-IC}, s_{LE-IC})$ is sent to the IC.

Step 2:

$$(ID_{LE}, M_{LE-IC}, Cert_{LE}, AB_{LE}, TS_{LE-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{LE-IC})$$ (32)

is first calculated by the IC, $TS_{NOW} - TS_{LE-IC} \leq \Delta T$ is used to confirm whether the timestamp is valid, $Cert_{LE}$ is verified with $PK_{LE}$, and then the correctness of the ECDSA signature is verified.

$$token = h(ID_{LE}, M_{LE-IC}, Cert_{LE}, AB_{LE}, TS_{LE-IC}, ID_{BC})$$ (33)

is calculated, the verification process by $(token, r_{LE-IC}, s_{LE-IC}, Q_{LE})$ is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from the LE will be received by the IC, and the smart contracts LE_ICins and LE_Icchk are triggered. Algorithm 7 is presented as follows:

$$BC_{LE-IC} = (r_{LE-IC}, s_{LE-IC})$$ (34)

Is calculated by the IC, and $(ID_{BC}, BC_{LE-IC})$ will also be uploaded to the blockchain center. Then, a random value $k_{IC-LE}$ is generated by the IC, and

$$token = h(ID_{IC}, M_{IC-LE}, Cert_{IC}, IC_{TID}, IC_{LE}, TS_{IC-LE}, ID_{BC})$$ (35)

is calculated. The signature process by $(token, k_{IC-LE}, d_{IC})$ is called, $(r_{IC-LE}, s_{IC-LE})$ is obtained,
\[ Enc_{IC-LE} = E_{PK_{LE}}(ID_{IC}, M_{IC-LE}, Cert_{IC}, IC_{TID}, IC_{LE}, TS_{IC-LE}, ID_{BC}) \]  

is calculated, the CA communication process is called, and \( ID_{IC}, Enc_{IC-LE}, (r_{IC-LE}, s_{IC-LE}) \) is sent to LE.

**Algorithm 7.** The proposed scheme’s smart contract LE_Icins and LE_Icchk

```plaintext
func ins SC LE_Icins (str Leid, str Ledetail, str Lecert, str Letsp) {
    cnt ++;
    LE[cnt].id = id;
    LE[cnt].detail = detail;
    LE[cnt].cert = cert;
    LE[cnt].tsp = tsp;
}
sign str Lekey (Leid, Ledetail, Lecert, Letsp);
verify str Lekey (Leid, Ledetail, Lecert, Letsp);

func chk SC LE_Icchk (str Leid, str Ledetail, str Lecert, str Letsp) {
    rtn Leid.exist;
    rtn Ledetail.exist;
    rtn Lecert.exist;
    rtn Letsp.exist;
}
```

Step 3:  
\[ (ID_{IC}, M_{IC-LE}, Cert_{IC}, IC_{TID}, IC_{LE}, TS_{IC-LE}, ID_{BC}) = D_{SK_{LE}}(Enc_{IC-LE}) \]  

is first calculated by the LE, \( TS_{NOW} - TS_{IC-LE} \leq \Delta T \) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified.

\[ token = h(ID_{IC}, M_{IC-LE}, Cert_{IC}, IC_{TID}, IC_{LE}, TS_{IC-LE}, ID_{BC}) \]  

is calculated, the verification process by \( (token, r_{IC-LE}, s_{IC-LE}, Q_{IC}) \) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from the IC will be received by LE, and the smart contracts IC_LEins and IC_Lechk will be triggered. Algorithm 8 is presented as follows:

**Algorithm 8.** The proposed scheme’s smart contract IC_Leins and IC_Lechk

```plaintext
func ins SC IC_Leins (str Icid, str Icdetail, str Iccert, str IC_LEICtid, str Ictsp) {
    cnt ++;
    IC[cnt].id = id;
    IC[cnt].detail = detail;
    IC[cnt].cert = cert;
    IC[cnt].LEICtid = LEICtid;
    IC[cnt].tsp = tsp;
}
sign str Ickey (Icid, Icdetail, Iccert, IC_LEICtid, Ictsp);
verify str Ickey (Icid, Icdetail, Iccert, IC_LEICtid, Ictsp);

func chk SC IC_Lechk (str Icid, str Icdetail, str Iccert, str IC_LEICtid, str Ictsp) {
    rtn Icid.exist;
    rtn Icdetail.exist;
    rtn Iccert.exist;
    rtn IC_LEICtid.exist;
    rtn Ictsp.exist;
}
```

BC_{IC-LE} = (r_{IC-LE}, s_{IC-LE})  

is calculated by LE, and \( (ID_{BC}, BC_{IC-LE}) \) will also be uploaded to the blockchain center.
3.9. Bank Payment Phase

After the licensee (LE) and art bank (AB) negotiate an artwork rental contract, the LE also completes the insurance of the artwork with the insurance company (IC). The LE then pays through the Bank (BK), which includes the rent paid to the AB and the insurance fee paid to the IC. The BK will send the relevant payment receipt back to the LE. We present the flowchart of the bank payment phase in Figure 8.

Step 1: A random value $k_{LE-BK}$ is generated by the LE,

$$\text{token} = h(ID_{LE}, M_{LE-BK}, Cert_{LE}, BK_{LE}, AB_{TID}, IC_{TID}, TS_{LE-BK}, ID_{BC})$$

(40)
is calculated, the signature process by $(token, k_{LE-BK}, d_{LE})$ is called, $(r_{LE-BK}, s_{LE-BK})$ is obtained,

$$Enc_{LE-BK} = E_{PK_{BK}}(ID_{LE}, M_{LE-BK}, Cert_{LE}, BK_{LE}, AB_{TID}, IC_{TID}, TS_{LE-BK}, ID_{BC})$$

(41)
is calculated, and $ID_{LE}, Enc_{LE-BK}, (r_{LE-BK}, s_{LE-BK})$ is sent to the BK.

Step 2:

$$(ID_{LE}, M_{LE-BK}, Cert_{LE}, BK_{LE}, AB_{TID}, IC_{TID}, TS_{LE-BK}, ID_{BC}) = D_{SK_{BK}}(Enc_{LE-BK})$$  \hspace{1cm} (42)

is first calculated by the BK, $TS_{NOW} - TS_{LE-BK} \leq \Delta T$ is used to confirm whether the timestamp is valid, $Cert_{LE}$ is verified with $PK_{LE}$, and then the correctness of the ECDSA signature is verified.

$$token = h(ID_{LE}, M_{LE-BK}, Cert_{LE}, BK_{LE}, AB_{TID}, IC_{TID}, TS_{LE-BK}, ID_{BC})$$  \hspace{1cm} (43)

is calculated, the verification process by $(token, r_{LE-BK}, s_{LE-BK}, Q_{LE})$ is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from the LE will be received by the BK, and the smart contracts LE_BKins and LE_BKchk are triggered. Algorithm 9 is presented as follows:

$$BC_{LE-BK} = (r_{LE-BK}, s_{LE-BK})$$  \hspace{1cm} (44)

is calculated by the BK, and $(ID_{BC}, BC_{LE-BK})$ will also be uploaded to the blockchain center. Then, a random value $k_{BK-LE}$ is generated by the BK, and

$$token = h(ID_{BK}, M_{BK-LE}, Cert_{BK}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{BK-LE}, ID_{BC})$$  \hspace{1cm} (45)

is calculated, the signature process by $(token, k_{BK-LE}, d_{BK})$ is called, $(r_{BK-LE}, s_{BK-LE})$ is got,

$$Enc_{BK-LE} = E_{PK_{LE}}(ID_{BK}, M_{BK-LE}, Cert_{BK}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{BK-LE}, ID_{BC})$$  \hspace{1cm} (46)

is calculated, the CA communication process is called, and $ID_{BK}, Enc_{BK-LE}, (r_{BK-LE}, s_{BK-LE})$ is sent to the LE.

**Algorithm 9.** The proposed scheme’s smart contract LE_BKins and LE_BKchk

```plaintext
func ins SC LE_BKins (str LEid, str LEdetail, str LEcert, str LE_LEABtid, str LE_LEICtid, str LEtsp) {
    cnt ++;
    LE[cnt].id = id;
    LE[cnt].detail = detail;
    LE[cnt].cert = cert;
    LE[cnt].LEABtid = LEABtid;
    LE[cnt].LEICtid = LEICtid;
    LE[cnt].tsp = tsp;
}

sign str LEkey (LEid, LEdetail, LEcert, str LE_LEABtid, str LE_LEICtid, str LEtsp);
verify str LEkey (LEid, LEdetail, LEcert, LE_LEABtid, LE_LEICtid, LEtsp);
func chk SC LE_BKchk (str LEid, str LEdetail, str LEcert, str LE_LEABtid, str LE_LEICtid, str LEtsp) {
    rtn LEid.exist;
    rtn LEdetail.exist;
    rtn LEcert.exist;
    rtn LE_LEABtid.exist;
    rtn LE_LEICtid.exist;
    rtn LEtsp.exist;
}
```
Figure 8. Bank payment phase.

Step 3:

\[
(ID_{BK}, M_{BK-L\text{E}}, Cert_{BK}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{BK-L\text{E}}, ID_{BK}) = D_{SK_{LE}}(Enc_{BK-L\text{E}}) \tag{47}
\]

is first calculated by the LE, \(TS_{NOW} - TS_{BK-L\text{E}} \leq \Delta T\) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified.

\[
token = h(ID_{BK}, M_{BK-L\text{E}}, Cert_{BK}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{BK-L\text{E}}, ID_{BK}) \tag{48}
\]

is calculated, the verification process by \((token, r_{BK-L\text{E}}, s_{BK-L\text{E}}, Q_{BK})\) is called, and a result of valid/invalid is obtained. If the verification is passed, the message from the BK will be received by the LE, and the smart contracts BK_LEins and BK_LEchk will be triggered. Algorithm 10 is presented as follows:
\[ BC_{BK-LE} = (r_{BK-LE}, s_{BK-LE}) \] (49)

is calculated by LE, and \((ID_{BC}, BC_{BK-LE})\) will also be uploaded to the blockchain center.

### 3.10. Contract Fulfillment Phase

After the licensee (LE) pays through the Bank (BK), the relevant payment receipt from the BK is obtained. The LE then informs the Art Bank (AB) that the rent and insurance fees have been paid, and provides relevant payment receipts. The AB verifies whether the payment receipt provided by the LE is consistent with the information in the Blockchain Center, and if it is confirmed, the artwork will be rented in accordance with the licensee’s requirements. We present the flowchart of the bank payment phase in Figure 9.

**Algorithm 10.** The proposed scheme’s smart contract BK_LEins and BK_LEchk

```
func ins SC BK_LEins (str BKid, str BKdetail, str BKcert, str BK_LErt, str BKtsp) {
    cnt ++;
    BK[cnt].id = id;
    BK[cnt].detail = detail;
    BK[cnt].cert = cert;
    BK[cnt].LErt = LLErt;
    BK[cnt].tsp = tsp;
}

sign str BKkey (BKid, BKdetail, BKcert, BK_LErt, BKtsp);
verify str BKkey (BKid, BKdetail, BKcert, BK_LErt, BKtsp);

func chk SC BK_LEchk (str BKid, str BKdetail, str BKcert, str BK_LErt, str BKtsp) {
    rtn BKid.exist;
    rtn BKdetail.exist;
    rtn BKcert.exist;
    rtn BK_LErt.exist;
    rtn BKtsp.exist;
}
```

**Step 1:** A random value \(k_{LE-AB}\) is generated by the LE,

\[
\text{token} = h(ID_{LE}, M_{LE-AB}, Cert_{LE}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{LE-AB}, ID_{BC})
\] (50)

is calculated, the signature process by \((\text{token}, k_{LE-AB}, d_{LE})\) is called, \((r_{LE-AB}, s_{LE-AB})\) is got,

\[
Enc_{LE-AB} = E_{PK_{AB}}(ID_{LE}, M_{LE-AB}, Cert_{LE}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{LE-AB}, ID_{BC})
\] (51)

is calculated, and \(ID_{LE}, Enc_{LE-AB}, (r_{LE-AB}, s_{LE-AB})\) is sent to the AB.

**Step 2:**

\[
(ID_{LE}, M_{LE-AB}, Cert_{LE}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{LE-AB}, ID_{BC}) = D_{SK_{AB}}(Enc_{LE-AB})
\] (52)

is first calculated by the AB, \(TS_{NOW} - TS_{LE-AB} \leq \Delta T\) is used to confirm whether the timestamp is valid, \(Cert_{LE}\) is verified with \(PK_{LE}\), and then the correctness of the ECDSA signature is verified.

\[
\text{token} = h(ID_{LE}, M_{LE-AB}, Cert_{LE}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{LE-AB}, ID_{BC})
\] (53)

is calculated, the verification process by \((\text{token}, r_{LE-AB}, s_{LE-AB}, Q_{LE})\) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from the LE will be received by the AB, and the smart contracts LE_ABIns and LE_ABchk are triggered.

**Algorithm 11 is presented as follows:**
is calculated by the AB, and \((ID_{BC}, BC_{LE-AB})\) will also be uploaded to the blockchain center. Then, a random value \(k_{AB-LE}\) is generated by the AB, and

\[
\text{token} = \hat{h}(ID_{AB}, M_{AB-LE}, Cert_{AB}, TS_{AB-LE}, ID_{BC})
\]

(55)
is calculated, the signature process is called by \((\text{token}, k_{AB-LE}, d_{AB})\), \((r_{AB-LE}, s_{AB-LE})\) is obtained,

\[
Enc_{AB-LE} = E_{PK_{LE}}(ID_{AB}, M_{AB-LE}, Cert_{AB}, TS_{AB-LE}, ID_{BC})
\]

(56)
is calculated, the CA communication process is called, and then \(ID_{AB}, Enc_{AB-LE}, (r_{AB-LE}, s_{AB-LE})\) is sent to the LE.

Step 3: \((ID_{AB}, M_{AB-LE}, Cert_{AB}, TS_{AB-LE}, ID_{BC}) = D_{SK_{LE}}(Enc_{AB-LE})\)

(57)
is first calculated by the LE, \(TS_{NOW} - TS_{AB-LE} \leq \Delta T\) is used to confirm whether the timestamp is valid, and then the correctness of the ECDSA signature is verified,

\[
\text{token} = \hat{h}(ID_{AB}, M_{AB-LE}, Cert_{AB}, TS_{AB-LE}, ID_{BC})
\]

(58)
is calculated, the verification process by \((\text{token}, r_{AB-LE}, s_{AB-LE}, Q_{AB})\) is called, and then a result of valid/invalid is obtained. If the verification is passed, the message from the AB will be received by the LE, and the smart contracts \(AB_{LEins}\) and \(AB_{LEchk}\) will be triggered. Algorithm 12 is presented as follows:

**Algorithm 11.** The proposed scheme’s smart contract \(LE_{ABins}\) and \(LE_{ABchk}\)

```c
func ins SC LE_ABins (str LEid, str LEdetail, str LEcert, str LE_LErt, str LE_LEABtid, str LE_LEICtid, str LEtsp) {
    cnt ++;
    LE[cnt].id = id;
    LE[cnt].detail = detail;
    LE[cnt].cert = cert;
    LE[cnt].LErt = LErt;
    LE[cnt].LEABtid = LEABtid;
    LE[cnt]. LEICtid = LEICtid;
    LE[cnt].tsp = tsp;
}

sign str LEk ery (LEid, LEdetail, LEcert, LE_LErt, LE_LEABtid, LE_LEICtid, LEtsp);
verify str LEk ekey (LEid, LEdetail, LEcert, LE_LErt, LE_LEABtid, LE_LEICtid, LEtsp);
func chk SC LE_ABchk (str LEid, str LEdetail, str LEcert, str LE_LErt, str LE_LEABtid, str LE_LEICtid, str LEtsp) {
    rtn LEid.exist;
    rtn LEdetail.exist;
    rtn LEcert.exist;
    rtn LE_LErt.exist;
    rtn LE_LEABtid.exist;
    rtn LE_LEICtid.exist;
    rtn LEtsp.exist;
}
```
Figure 9. Contract fulfilment phase.

\[ BC_{AB-LB} = (r_{AB-LB}, s_{AB-LB}) \]  

is calculated by the LE, and \( (ID_{BC}, BC_{AB-LB}) \) will also be uploaded to the blockchain center.
Algorithm 12. The proposed scheme’s smart contract AB_LEIns and AB_LEchk

```c
func ins SC AB_LEIns (str ABid, str ABdetail, str ABcert, str ABtsp) {
    cnt ++;
    AB[cnt].id = id;
    AB[cnt].detail = detail;
    AB[cnt].cert = cert;
    AB[cnt].tsp = tsp;
}

sign str ABkey (ABid, ABdetail, ABcert, ABtsp);
verify str ABkey (ABid, ABdetail, ABcert, ABtsp);
func chk SC AB_LEchk (str ABid, str ABdetail, str ABcert, str ABtsp) {
    rtn ABid.exist;
    rtn ABdetail.exist;
    rtn ABcert.exist;
    rtn ABtsp.exist;
}
```

3.11. Arbitration Mechanism Phase

When there is an argument between the AB, IC, BK, and LE, disputes can be resolved through arbitration. The LE will have appealed to the AI and obtained permission to lease the artwork from the AB, who will have obtained insurance from the IC, and paid the rent and premium, but AB did not provide the artwork as agreed. When the LE requests arbitration from the AI, the AI compares blockchain messages through the signature information to confirm the lease contract. We present the arbitration mechanism verification phase in Figure 10.

Step 1: The certificate, signature, and blockchain data of the package through $ID_{BC}$ can all be downloaded by the AI, and then

$$BC_{LE-AB} \equiv (r_{LE-AB}, s_{LE-AB})$$

and

$$BC_{AB-LE} \equiv (r_{AB-LE}, s_{AB-LE})$$

are verified by the AI. If the verification fails, there is no lease application from the LE to the AB, so AB does not need to provide artwork to the LE.

Step 2: If the signature between the LE and AB, and the blockchain data are all verified and valid, then

$$BC_{LE-IC} \equiv (r_{LE-IC}, s_{LE-IC})$$

and

$$BC_{IC-LE} \equiv (r_{IC-LE}, s_{IC-LE})$$

will be verified by the AI. If the verification fails, the LE is shown to not have purchased insurance from the IC. After the AB comprehensively assesses LE’s financial and credit status and the value of the artwork, the AB has the right to reject the application for the lease of artwork that is not insured. If the LE has paid the rent, the AB shall refund the rent to the LE.

Step 3: If the signature between the LE and the IC, and the blockchain data are all verified and valid, then

$$BC_{LE-BK} \equiv (r_{LE-BK}, s_{LE-BK})$$

and

$$BC_{BK-LE} \equiv (r_{BK-LE}, s_{BK-LE})$$

will be verified by the AI. If the verification fails, the LE has applied to the AB for the lease of artwork, and has also applied for insurance from the IC, but has not paid the rent and premium, so the AB does not need to provide the artwork to the LE.
Step 4: If the signature between the LE and the BK, and the blockchain data are verified and valid, then the blockchain data are fully verified. The LE has applied to the AB for the lease of artwork, and has also applied for insurance from the IC, and completed the payment of rent and insurance. Therefore, the AB should provide the artwork to the LE in accordance with the lease.

![Diagram](image)

**Figure 10.** The arbitration mechanism verification phase.

4. Analysis of Advantage and Threat Model

4.1. Decentralization and Information Sharing

In our proposed architecture and communication protocol, messages transmitted in the system will be computed by each role and signed with the role’s private key. The blockchain environment of this study is also defined as the Hyperledger consortium chain environment. Under the framework of the Hyperledger consortium chain, all members in the blockchain must be registered before they can share information. For members in the same Hyperledger consortium chain, the transmission of all node information is open and transparent, that is, a single node in the alliance cannot deceive other nodes. Through such an operation mechanism, we will be able to realize the trust relationship between nodes. Therefore, the scheme proposed in this study has indeed realized the decentralization and information sharing of alliance members.
4.2. Openness

In the architecture and communication protocol proposed in this study, the public key cryptosystem is adopted. Through the public key cryptography system, combined with the operating environment of the consortium chain, information that can be publicly queried and verified can be generated on the chain. When applied to the art leasing environment, information related to art leasing will be open and transparent, reducing information asymmetry and reducing public distrust. In addition, through open and transparent open information, the related applications of big data and cloud computing can also be improved, and the price of art rental will be more reasonable and accurate.

4.3. Privacy and Access Control

This study adopts the alliance chain architecture of blockchain technology and has a public key cryptosystem and access control, which can fully protect the privacy of members. This framework is applied to the art leasing environment. The key art leasing contract will only be eligible to be viewed by both parties to the lease, and other members of the alliance will not be able to access these privacy-protected messages. The query and modification records of the art leasing contract will also be completely stored on the blockchain, which not only protects privacy but also satisfies the relevant characteristics of the blockchain.

4.4. Mutual Authentication

At each communication stage of our proposed scheme, the main purpose of the scheme is to authenticate the private keys of party X and party Y, which includes the Licensee (LE), Bank (BK), Insurance Company (IC), Art Bank (AB), and Certificate Authority (CA).

\begin{align*}
G1: & \quad X \equiv X^{x_{Y \to X}} \equiv Y \\
G2: & \quad X \equiv Y^{y_{X \to Y}} \equiv X \\
G3: & \quad Y \equiv X^{y_{X \to Y}} \equiv X \\
G4: & \quad Y \equiv X^{x_{Y \to X}} \equiv Y \\
G5: & \quad X \equiv ID_Y \\
G6: & \quad X \equiv Y \Rightarrow ID_Y \\
G7: & \quad Y \equiv ID_X \\
G8: & \quad Y \equiv X \Rightarrow ID_X
\end{align*}

Applying the BAN logic to the scheme proposed in this study, the following idealized form is derived:

\begin{align*}
M1: & \quad (< ID_X, k_{X \to Y}, TS_{X \to Y} >_{PK_Y}, < H(ID_X,k_{X \to Y},TS_{X \to Y}) >_{x_{X \to Y}}) \\
M2: & \quad (< ID_Y, k_{Y \to X}, TS_{Y \to X} >_{PK_X}, < H(ID_Y,k_{Y \to X},TS_{Y \to X}) >_{x_{Y \to X}})
\end{align*}

We analyze the proposed scheme by making the following assumptions:

\begin{align*}
A1: & \quad X \equiv \#(k_{X \to Y}) \\
A2: & \quad Y \equiv \#(k_{X \to Y}) \\
A3: & \quad X \equiv \#(k_{Y \to X}) \\
A4: & \quad Y \equiv \#(k_{Y \to X}) \\
A5: & \quad X \equiv Y \Rightarrow X^{x_{Y \to X}} \equiv Y \\
A6: & \quad Y \equiv X \Rightarrow X^{x_{X \to Y}} \equiv Y \\
A7: & \quad X \equiv Y \Rightarrow ID_Y \\
A8: & \quad Y \equiv X \Rightarrow ID_X
\end{align*}

According to the assumptions and rules of the BAN logic in each communication stage, the main proof is as follows:

a. Party Y confirms party X.
We conduct the following declaration through $M1$ and the seeing rule:
\[ Y \vartriangleleft (<ID_X,k_{X-Y},TS_{X-Y} \triangleright PK_Y, <H(ID_X,k_{X-Y},TS_{X-Y}) \triangleright s_{X-Y}) \] (66)

We conduct the following declaration through $A2$ and the freshness rule:
\[ Y \equiv (,<ID_X,k_{X-Y},TS_{X-Y} \triangleright PK_Y, <H(ID_X,k_{X-Y},TS_{X-Y}) \triangleright s_{X-Y}) \] (67)

We conduct the following declaration through (66), $A4$, and the message meaning rule:
\[ Y \equiv X \sim (<ID_X,k_{X-Y},TS_{X-Y} \triangleright PK_Y, <H(ID_X,k_{X-Y},TS_{X-Y}) \triangleright s_{X-Y}) \] (68)

We conduct the following declaration through (67), (68), and the nonce verification rule:
\[ Y \equiv X \equiv (<ID_X,k_{X-Y},TS_{X-Y} \triangleright PK_Y, <H(ID_X,k_{X-Y},TS_{X-Y}) \triangleright s_{X-Y}) \] (69)

We conduct the following declaration through (70) and the belief rule:
\[ Y \equiv X \equiv X^x_{x^y} Y \] (70)

We conduct the following declaration through (70), $A6$, and the jurisdiction rule:
\[ Y \equiv X \equiv X^x_{x^y} Y \] (71)

We conduct the following declaration through (71) and the belief rule:
\[ Y \equiv X \equiv ID_X \] (72)

We conduct the following declaration through (72), $A8$, and the jurisdiction rule:
\[ Y \equiv ID_X \] (73)

b. Party X confirms party Y.

We conduct the following declaration through $M2$ and the seeing rule:
\[ X \vartriangleleft (<ID_Y,k_{Y-X},TS_{Y-X} \triangleright PK_X, <H(ID_Y,k_{Y-X},TS_{Y-X}) \triangleright s_{Y-X}) \] (74)

We conduct the following declaration through $A1$ and the freshness rule:
\[ X \equiv (,<ID_Y,k_{Y-X},TS_{Y-X} \triangleright PK_X, <H(ID_Y,k_{Y-X},TS_{Y-X}) \triangleright s_{Y-X}) \] (75)

We conduct the following declaration through (74), $A3$, and the message meaning rule:
\[ X \equiv Y \sim (<ID_Y,k_{Y-X},TS_{Y-X} \triangleright PK_X, <H(ID_Y,k_{Y-X},TS_{Y-X}) \triangleright s_{Y-X}) \] (76)

We conduct the following declaration through (75), (76), and the nonce verification rule:
\[ X \equiv Y \equiv (<ID_Y,k_{Y-X},TS_{Y-X} \triangleright PK_X, <H(ID_Y,k_{Y-X},TS_{Y-X}) \triangleright s_{Y-X}) \] (77)

We conduct the following declaration through (77) and the belief rule:
\[ X \equiv Y \equiv X^y_{x^y} Y \] (78)

We conduct the following declaration through (78), $A5$, and the jurisdiction rule:
\[ X \equiv X^y_{x^y} Y \] (79)
We conduct the following declaration through (79) and the belief rule:

\[ X|\equiv Y|\equiv ID_Y \]  

We conduct the following declaration through (80), A7, and the jurisdiction rule:

\[ X|\equiv ID_Y \]  

In our proposed scheme, party X and party Y authentication has been proved through (71), (73), (79), and (81). Furthermore, the proposed scheme can verify the private keys of party X and party Y, which is also provable.

In our proposed scheme, we choose the artwork rental phase for example. The licensee is authenticated by the art bank through \( x_{LE-AB} \equiv r_{LE-AB} \mod n \). The legality of the licensee is confirmed by the art bank if the verification has been passed. The art bank is authenticated by the licensee through \( x_{AB-LE} \equiv r_{AB-LE} \mod n \). The legality of the art bank is confirmed by the licensee if the verification has been passed. Mutual authentication between the art bank and the licensee is guaranteed in the artwork rental phase of our proposed scheme.

4.5. Verifiable

Digital certificate verification can be publicly used to verify the identity of each role, and authorization information is also public. The openness and transparency of blockchain information can make the art rental field professional and efficient.

Let’s take the message transmitted by the LE and IC as an example. When LE sends a message signed by ECDSA to the IC, the IC first verifies the correctness of the timestamp and signature, then generates blockchain data \( BC_{LE-IC} = (t_{PT-IC}, z_{PT-IC}) \), and uses \( ID_{BC} \) as an index to upload the blockchain data to the Certificate Authority. In other words, after verifying the correctness of the timestamp and signature of each role receiving the message, the correctness of the blockchain data generated by the previous role is also verified. Thus, the proposed method achieves the characteristics of public verification through the technology of blockchain and the digital signature of ECDSA.

4.6. Integrity and Unforgery

Using a timestamp and signature mechanism, a string of random numbers and letters is irreversibly generated for the data placed in each block. The original text cannot be inferred from the string, which effectively solves the trust problem. After the hash function operation, the message is described as follows.

\[
\begin{align*}
token & = h(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC}) \\
token & = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC}) \\
token & = h(ID_{LE}, M_{LE-AB}, Cert_{LE}, TS_{LE-AB}, ID_{BC}) \\
token & = h(ID_{AB}, M_{AB-LE}, Cert_{AB}, AB_{TID}, AB_{LE}, TS_{AB-LE}, ID_{BC}) \\
token & = h(ID_{LE}, M_{LE-IC}, Cert_{LE}, AB_{LE}, TS_{LE-IC}, ID_{BC}) \\
token & = h(ID_{IC}, M_{IC-LE}, Cert_{IC}, IC_{TID}, IC_{LE}, TS_{IC-LE}, ID_{BC}) \\
token & = h(ID_{LE}, M_{LE-BK}, Cert_{LE}, BK_{LE}, AB_{TID}, IC_{TID}, TS_{LE-BK}, ID_{BC}) \\
token & = h(ID_{BK}, M_{BK-LE}, Cert_{BK}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{BK-LE}, ID_{BC}) \\
token & = h(ID_{LE}, M_{LE-AB}, Cert_{LE}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{LE-AB}, ID_{BC}) \\
token & = h(ID_{AB}, M_{AB-LE}, Cert_{AB}, TS_{AB-LE}, ID_{BC})
\end{align*}
\]

The hash value cannot be reversely restored to the original content, so this protocol implements the feature that the message cannot be tampered with.

4.7. Traceable

In the blockchain technology combined with the alliance chain architecture proposed in this research, when the transmitted message is uploaded to the blockchain through
the CA, the message will be permanently stored on the blockchain and cannot be tampered with, so it can be applied to follow-up tracking. For example: when we want to verify and trace whether the blockchain data between the LE and BK is legal in the bank payment phase, we can compare and verify $BC_{LE-BK} = (r_{LE-BK}, s_{LE-BK})$ and $BC_{BK-LE} = (r_{BK-LE}, s_{BK-LE})$. When we want to verify and trace whether the blockchain data between the LE and AB is legal in the contract fulfillment phase, we can compare and verify $BC_{LE-AB} = (r_{LE-AB}, s_{LE-AB})$ and $BC_{AB-LE} = (r_{AB-LE}, s_{AB-LE})$.

4.8. Non-repudiation

The framework and communication protocol proposed in this research adopts the public key cryptosystem. The sender of the message in the system signs the message with his/her private key. After receiving the message, the receiver of the message will use the sender’s public key to authenticate the message. If the receiver successfully authenticates the message sent by the sender, the sender can no longer deny the message. These messages will also be uploaded to the blockchain for permanent storage. If an attacker wants to modify the content, he/she needs to control more than half of the system nodes, which will be very difficult. An undeniable description of each role in the proposed scheme is shown in the following Table 1.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Item</th>
<th>Signature</th>
<th>Sender</th>
<th>Receiver</th>
<th>Signature Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA communication process</td>
<td>$(r_{AP-CA}, s_{AP-CA})$</td>
<td>AP</td>
<td>CA</td>
<td></td>
<td>$x_{AP-CA} \equiv r_{AP-CA} \mod n$</td>
</tr>
<tr>
<td></td>
<td>$(r_{CA-AP}, s_{CA-AP})$</td>
<td>CA</td>
<td>AP</td>
<td></td>
<td>$x_{CA-AP} \equiv r_{CA-AP} \mod n$</td>
</tr>
<tr>
<td>Artwork rental phase</td>
<td>$(r_{LE-AB}, s_{LE-AB})$</td>
<td>LE</td>
<td>AB</td>
<td></td>
<td>$x_{LE-AB} \equiv r_{LE-AB} \mod n$</td>
</tr>
<tr>
<td></td>
<td>$(r_{AB-LE}, s_{AB-LE})$</td>
<td>AB</td>
<td>LE</td>
<td></td>
<td>$x_{AB-LE} \equiv r_{AB-LE} \mod n$</td>
</tr>
<tr>
<td>Insurance insured phase</td>
<td>$(r_{LE-IC}, s_{LE-IC})$</td>
<td>LE</td>
<td>IC</td>
<td></td>
<td>$x_{LE-IC} \equiv r_{LE-IC} \mod n$</td>
</tr>
<tr>
<td></td>
<td>$(r_{IC-LE}, s_{IC-LE})$</td>
<td>IC</td>
<td>LE</td>
<td></td>
<td>$x_{IC-LE} \equiv r_{IC-LE} \mod n$</td>
</tr>
<tr>
<td>Bank payment phase</td>
<td>$(r_{LE-BK}, s_{LE-BK})$</td>
<td>LE</td>
<td>BK</td>
<td></td>
<td>$x_{LE-BK} \equiv r_{LE-BK} \mod n$</td>
</tr>
<tr>
<td></td>
<td>$(r_{BK-LE}, s_{BK-LE})$</td>
<td>BK</td>
<td>LE</td>
<td></td>
<td>$x_{BK-LE} \equiv r_{BK-LE} \mod n$</td>
</tr>
<tr>
<td>Contract fulfillment phase</td>
<td>$(r_{LE-AB}, s_{LE-AB})$</td>
<td>LE</td>
<td>AB</td>
<td></td>
<td>$x_{LE-AB} \equiv r_{LE-AB} \mod n$</td>
</tr>
<tr>
<td></td>
<td>$(r_{AB-LE}, s_{AB-LE})$</td>
<td>AB</td>
<td>LE</td>
<td></td>
<td>$x_{AB-LE} \equiv r_{AB-LE} \mod n$</td>
</tr>
</tbody>
</table>

4.9. Resist Man-in-the-Middle Attack

In the architecture and communication protocol proposed in this study, the public key cryptosystem is adopted, which can effectively prevent man-in-the-middle attacks. All the content transmitted by the sender will be signed with its private key and encrypted with the receiver’s public key. When a malicious person modifies the content transmitted by the sender, because the sender’s private key cannot be used to sign it, the receiver will judge it as an invalid message and successfully block the man-in-the-middle attack.

Scenario: The content from the sender to the receiver is intercepted and tampered with by the attacker before sending it to the receiver.

Analysis: The attacker will fail due to the proposed solution using a public key to encrypt content as follows.

$$Enc_{AP-CA} = Enc_{PK_CA}(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC})$$

$$Enc_{CA-AP} = Enc_{PK_AP}(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC})$$

$$Enc_{LE-AB} = Enc_{PK_LE}(ID_{LE}, M_{LE-AB}, Cert_{LE}, TS_{LE-AB}, ID_{BC})$$

$$Enc_{AB-LE} = Enc_{PK_LE}(ID_{AB}, M_{AB-LE}, Cert_{AB}, AB_{TID}, AB_{LE}, TS_{AB-LE}, ID_{BC})$$
\[ Enc_{LE-IC} = E_{PK_{IC}}(ID_{LE}, M_{LE-IC}, Cert_{LE}, AB_{LE}, TS_{LE-IC}, ID_{BC}) \]
\[ Enc_{IC-LE} = E_{PK_{LE}}(ID_{IC}, M_{IC-LE}, Cert_{IC}, IC_{TID}, IC_{LE}, TS_{IC-LE}, ID_{BC}) \]
\[ Enc_{LE-BK} = E_{PK_{BK}}(ID_{LE}, M_{LE-BK}, Cert_{LE}, BK_{LE}, AB_{TID}, IC_{TID}, TS_{LE-BK}, ID_{BC}) \]
\[ Enc_{BK-LE} = E_{PK_{LE}}(ID_{BK}, M_{BK-LE}, Cert_{BK}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{BK-LE}, ID_{BC}) \]
\[ Enc_{LE-AB} = E_{PK_{AB}}(ID_{LE}, M_{LE-AB}, Cert_{LE}, RT_{LE}, AB_{TID}, IC_{TID}, TS_{LE-AB}, ID_{BC}) \]
\[ Enc_{AB-LE} = E_{PK_{LE}}(ID_{AB}, M_{AB-LE}, Cert_{AB}, TS_{AB-LE}, ID_{BC}) \]

Malicious attackers do not have the correct private key; thus, they will not be able to obtain the encrypted data. The attacker cannot decrypt and tamper with the message even if he/she intercepts the message. Thus, the attacker cannot modify the message and send it to the receiver in this study.

5. Discussions

5.1. Computation Cost Analysis

In Table 2, the computational costs of each phase are analyzed. The asymmetrical, comparison, hash function, and multiplication operations are applied as the basis for calculating the cost.

**Table 2. Computation costs of the proposed scheme.**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Party</th>
<th>LE</th>
<th>AB</th>
<th>IC</th>
<th>BK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artwork rental phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Insurance insured phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>N/A</td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>N/A</td>
</tr>
<tr>
<td>Bank payment phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>N/A</td>
<td>N/A</td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
</tr>
<tr>
<td>Contract fulfillment phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>2T_{sig} + 2T_{mp} + 2T_{h} + 7T_{mul}</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Notes: \(T_{sig}\): An asymmetrical signature/verifying a signature’s time is required. \(T_{mp}\): A comparison of the operation’s time is required. \(T_{h}\): A one-way hash function’s time is required. \(T_{mul}\): A multiplication operation’s time is required.

5.2. Communication Performance Analysis

In Table 3, the communication performance at each stage is analyzed. The maximum transmission speed is 100 Mbps in a 4G environment, while the maximum transmission speed is 20 Gbps [43] in a 5G environment.

**Table 3. Communication cost of the proposed scheme.**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Party</th>
<th>Message Length</th>
<th>Round</th>
<th>4G (100 Mbps)</th>
<th>5G (20 Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA communication process</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 5T_{alter} = 2<em>384 + 2</em>1024 + 2*160 = 3136 bits</td>
<td>2</td>
<td>3136/102,400 = 0.031 ms</td>
<td>3136/20,480,000 = 0.15 us</td>
</tr>
<tr>
<td>Artwork rental phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 5T_{alter} = 2<em>384 + 2</em>1024 + 2*160 = 3136 bits</td>
<td>2</td>
<td>3136/102,400 = 0.031 ms</td>
<td>3136/20,480,000 = 0.15 us</td>
</tr>
<tr>
<td>Insurance insured phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 5T_{alter} = 2<em>384 + 2</em>1024 + 2*160 = 3136 bits</td>
<td>2</td>
<td>3136/102,400 = 0.031 ms</td>
<td>3136/20,480,000 = 0.15 us</td>
</tr>
<tr>
<td>Bank payment phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 5T_{alter} = 2<em>384 + 2</em>1024 + 2*160 = 3136 bits</td>
<td>2</td>
<td>3136/102,400 = 0.031 ms</td>
<td>3136/20,480,000 = 0.15 us</td>
</tr>
<tr>
<td>Contract fulfillment phase</td>
<td></td>
<td>2T_{sig} + 2T_{mp} + 5T_{alter} = 2<em>384 + 2</em>1024 + 2*160 = 3136 bits</td>
<td>2</td>
<td>3136/102,400 = 0.031 ms</td>
<td>3136/20,480,000 = 0.15 us</td>
</tr>
</tbody>
</table>

Notes: \(T_{sig}\): To transmit an ECDSA signature’s (384 bits) time required. \(T_{mp}\): To transmit an asymmetric message’s (1024 bits) time required. \(T_{alter}\): To transmit information’s (160 bits) time required.
5.3. Efficiency Quantitative Analysis and Estimation

We provide two other research papers by other scholars, one of which conducts quantitative analysis of transactions per second and latency time between Ethereum 1.5.8 and Hyperledger Fabric 0.6, and the other conducts the quantitative analysis of transactions per second and latency time between Hyperledger Fabric 0.6 and Hyperledger Fabric 1.0. The efficiency quantitative analysis and estimation surveys are in Table 4.

Table 4. Efficiency quantitative analysis and estimation surveys.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Item</th>
<th>System</th>
<th>Transactions per Block/Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pongnumkul et al.</td>
<td>2017</td>
<td>Throughput</td>
<td>Ethereum 1.5.8</td>
<td>27.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hyperledger Fabric 0.6</td>
<td>68.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Latency</td>
<td>Ethereum 1.5.8</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hyperledger Fabric 0.6</td>
<td>0.11</td>
</tr>
<tr>
<td>Nasir et al.</td>
<td>2018</td>
<td>Throughput</td>
<td>Hyperledger Fabric 0.6</td>
<td>116.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hyperledger Fabric 1.0</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Latency</td>
<td>Hyperledger Fabric 0.6</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hyperledger Fabric 1.0</td>
<td>0.022</td>
</tr>
</tbody>
</table>

According to the performance analysis of 5.1 to 5.3 above, our system will adopt the Hyperledger Fabric 2.0 environment, of which the performance is expected to be greatly improved.

5.4. Characteristic Comparison

In this section, our proposed scheme and existing commerce-related decentralized blockchain surveys are compared in Table 5.

Table 5. Comparison of the proposed and existing commerce-related decentralized blockchain surveys.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Objective</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. [19]</td>
<td>2020</td>
<td>Proposed a blockchain-based intelligent anti-switch package in tracing logistics system</td>
<td>Y</td>
<td>NA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Xue et al. [12]</td>
<td>2021</td>
<td>Proposed a housing rental system based on blockchain technology</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
<td>NA</td>
</tr>
<tr>
<td>Our scheme</td>
<td>2022</td>
<td>Proposed an artwork rental system based on blockchain technology</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>


6. Conclusions and Future Work

In recent years, the rise of blockchain technology has completely changed people’s traditional life and thinking patterns, and has spawned many new blockchains, and business models. Blockchain technology will have a major impact and change in all walks of life, including leasing-related industries. This study proposes the complete application of blockchain technology in the field of art leasing, with blockchain technology expected to continue to shine at a higher level in the future.

Through this study, the following goals have been achieved:
(1) Lease items and prices are open and transparent: The art banks will form a consortium to jointly register and maintain the information on the artworks. The types of artwork and rental information are open and transparent. The renters can use reasonable rents to rent their favorite artworks, which will increase the renters’ willingness to rent.

(2) Public investment for mutual benefit and common prosperity: Usually, the price of art is extremely expensive. Art banks do not have enough funds to buy a large amount of art. At this time, people can be open to investing and subscribing to art. If the art is successfully rented out, the investing public can get profit dividends. Through the non-tampering feature of the blockchain and investors’ trust in the Art Bank is also improved.

(3) Positive circulation and sustainable operation: Through blockchain technology, renters can quickly find their favorite artworks due to the convenient search mechanism. Due to the open and transparent rental price, the renters’ willingness to rent will be greatly increased. Investors can check the lease status of the artwork they invest in at any time, and the dividend information is open and transparent, which improves investors’ willingness to invest. Such a positive cycle between the two parties will ensure the sustainable operation of the Art Bank.

Art banks can promote the awakening of art investment awareness and the prosperity of the art market, and the development of the art market will strengthen art investment awareness, thereby promoting the development of art banks. It can be expected that in this interactive relationship between the art bank and the art market, the art bank will go all the way.

The framework and protocols proposed in this article are not only applicable to art leasing, but also housing leasing, and even other leasing-related issues. However, all roles need to register in the blockchain center and obtain the public-private key pair of elliptic curve signatures to use the system in our proposed scheme. In addition, there needs to be an artwork information-sharing platform planned by the government or businesses in the environment.

Author Contributions: D.-C.H., L.-C.L., Y.-Y.D. and C.-L.C. contributed to the study’s conception and design. Material preparation, data collection, and analysis were performed by D.-C.H., L.-C.L., Y.-Y.D. and C.-L.C. The first draft of the manuscript was written by L.-C.L. and Y.-Y.D. and D.-C.H., L.-C.L., Y.-Y.D. and C.-L.C. commented on previous versions of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under contract MOST 111-2218-E-305-001–MBK and contract MOST 110-2410-H-324-004-MY2.

Institutional Review Board Statement: This study is only based on theoretical basic research. It is not involving humans.

Informed Consent Statement: This study is only based on theoretical basic research. It is not involving humans.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References
2. Chiou, S.-C.; Wang, Y.-C. The example application of a genetic algorithm for the framework of cultural and creative brand design in Tamsui Historical Museum. Soft Comput. 2018, 22, 2527–2545. [CrossRef]
Symmetry 2023, 15, 341


12. Xue, Q.; Hou, Z.; Ma, H.; Zhu, H.; Ju, X.; Sun, Y. Housing rental system based on blockchain Technology. IoTAIMA 2021, 1948, 012058. [CrossRef]


44. Uddin, M. Blockchain Medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry. Int. J. Pharm. 2021, 597, 120235. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.