symmetry

MDPI

Article

A Blockchain-Based Anti-Counterfeit and Traceable NBA Digital Trading Card Management System

Chin-Ling Chen [1,2], Cheng-Chen Fang [3], Ming Zhou [3], Woei-Jiunn Tsaur [4,5,*], Hongyu Sun [6,7,*], Wanbing Zhan [3] and Yong-Yuan Deng [2]

1 School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China
2 Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung City 413310, Taiwan
3 School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
4 Computer Center, National Taipei University, New Taipei City 237303, Taiwan
5 Department of Computer Science and Information Engineering, National Taipei University, New Taipei City 237303, Taiwan
6 Department of Computer Science, Jilin Normal University, Siping 136000, China
7 State Key Laboratory of Numerical Simulation, Siping 136000, China
* Correspondence: wjtsaur@mail.ntpu.edu.tw (W.-J.T.); hongyu@jlnu.edu.cn (H.S.)

Abstract: NBA (National Basketball Association) trading cards are a hot collector's item, with sales increasing rapidly every year. However, with the popularity of online trading, some sellers have started to intentionally and unintentionally sell imitation trading cards, and even PwC (Pricewaterhouse Coopers) is not immune. However, the PSA (Professional Sports Authenticator), which is the authentication agency, is not liable for this. Faced with the above situation, we moved trading cards online and proposed a blockchain-based anti-counterfeit and traceable NBA digital trading card management system, using blockchain technology to protect digital trading cards, and special digital copyright, to move from relying on other regulators to achieve the fight against counterfeit cards and maintain the security of the digital trading card market. Finally, we analyzed the security of the system and compared it with other methods. Our system uses Hyperledger Fabric to share data while protecting corporate privacy. Proxy re-encryption enables secure and trusted access authorization for digital transaction cards. Asymmetric encryption protects the data and uses signatures to achieve traceability and non-repudiation. Overall, our system solves the problem of counterfeiting and traceability that can occur in the digital trading card process from production to purchase.

Keywords: digital trading card; smart contract; digital rights management (DMR); Hyperledger Fabric blockchain; anti-counterfeiting; traceability

Citation: Chen, C.-L.; Fang, C.-C.; Zhou, M.; Tsaur, W.-J.; Sun, H.; Zhan, W.; Deng, Y.-Y. A Blockchain-Based Anti-Counterfeit and Traceable NBA Digital Trading Card Management System. *Symmetry* **2022**, *14*, 1827. https://doi.org/10.3390/sym14091827

Academic Editors: José Carlos R. Alcantud and Debiao He

Received: 24 July 2022
Accepted: 26 August 2022
Published: 2 September 2022

1. Introduction

1.1. Background

Trading cards are collectible cards, traditionally associated with sports, and sporting types of trading cards are also known as sports cards [1]. In the strong sports atmosphere in the United States, sports cards have become a widely popular trading card with the public. They usually include a picture of the player on one side and statistics or other information on the other. The images of athletes on sports cards and the unique player stories embedded in the cards make sports cards an integral part of sports culture. The global sports trading card market is valued at USD 13.82 billion in 2019 and is expected to reach USD 98.75 billion by 2027, growing at a CAGR (compound annual growth rate) of 23.01% from 2020 to 2027 [2]. The trading card craze is not limited to one category either. In 2020, eBay's trading card growth reached a record 162%, with total sales of basketball cards exploding by over 300% in the past year [3].

The explosion of trading cards has also attracted many outlaws. The Federal Bureau of Investigation launched Operation Bullpen, a sports memorabilia fraud-related operation, in the 1990s [4]. Experts estimate that counterfeit memorabilia accounts for more than USD 100 million of the market share each year. And with the popularity of eBay, unscrupulous dealers able to sell fake souvenirs without dealing face-to-face with customers have emerged. eBay's largest card seller, PwC (Pricewaterhouse Coopers), was also caught in a scandal, selling millions of dollars in altered cards [5]. These cards were graded by the PSA (Professional Sports Authenticator) [6] to estimate the value of the cards, but the PSA did not take any public responsibility. From the collector's point of view, the proliferation of counterfeit cards compromises the value of the cards, and collectors' trust in the PSA diminishes, but there is nothing they can do about it. There are obvious drawbacks to this centralized authentication mechanism and agency, and a more rational management system is needed to prevent the theft and counterfeiting of sports cards and to better regulate them.

The blockchain [7] is a game-changer. It allows platforms to track the ownership of digital assets (e.g., NFT—Non-Fungible Token) [8], making the records of these digital trading card transactions virtually unforgeable. If forgery does occur, thanks to traceability, those modified records are also recorded in the ledger, providing strong support for digital copyright protection. The decentralized nature of blockchain ensures the openness and transparency of information, equal power of bookkeeping, and multi-node storage, avoiding the risk of deception by authoritative centers and ensuring that the participants' information is equal. In summary, blockchain technology has a unique combination of a distributed ledger, decentralized structure, consensus algorithm, asymmetric encryption, and smart contracts. On the premise of retaining the portability and ornamental value of digital cards, it gives digital trading cards the same attribution and transaction value as physical trading cards, as well as better security, network visibility, and transparency than physical cards.

In summary, in this paper, we propose a blockchain-based anti-counterfeit traceable NBA digital trading card management system. The tamper-evident and decentralized nature of blockchain is utilized to ensure that the information of the uploaded digital transaction card is not tampered with. The property of blockchain technologies, such as the consensus mechanism, transparency, smart contract, and ECDSA (elliptic curve digital signature algorithm) are used to ensure the anti-counterfeit traceability of digital trading cards and enable the nodes to join the blockchain to reach trust and have secure and transparent access to information. Smart contracts are digitally written into each node of the blockchain, and then the smart contracts are executed by the blockchain to store and read data onto the blockchain. The blockchain provides a decentralized execution environment for smart contracts. Smart contracts make it easier for nodes in the blockchain to reach consensus, reducing costs and saving time for the blockchain. Blockchain and smart contracts complement each other [9,10]. Blockchain and smart contract technologies bring new solutions to several industry problems [11–14], such as combining ECDSA algorithms to curb counterfeiting in the pharmaceutical field [15]. This paper aims to protect the rights of NBA digital trading card collectors and maintain the economic health of the digital trading card market.

1.2. Related Works

A digital trading card is a kind of tradable digital content. For tradable digital content, its security involves the production, transaction, and authorization of digital content. Therefore, there is research on the security of digital content, the traceability of transactions, and the unforgeability of digital content [16–19]. NBA officials have cooperated with Dapper Lab to develop an application for NBA Top Shot on the flow blockchain platform to sell digital trading cards. However, the influx of users has caused them problems, and Dapper has now tried to limit the number of new users and the frequency of transactions to solve some of the problems associated with the influx of users [20]. Protection of digital content is an emerging area of blockchain application. These authors [21–26] reveal recent applications of blockchain, smart contracts, and cryptography for protecting digital content,

proposing a more specific and detailed approach. Table 1 compares existing surveys of digital content/digital rights management.

Table 1. Existing Digital Content/Digital Rights Management Survey.

Authors	Year	Objective	Technologies	Merits	Demerits
Ma et al. [21]	2018	Blockchain-based scheme for digital rights management	Ethereum, Smart Contract	Detailed implementation of DRM on a blockchain platform, transparency data format standardization	Timeless Not applicable
Ma et al. [22]	2019	Permissioned blockchain-based decentralized trust management and secure usage control scheme of IoT big data	Ethereum, Smart Contract	Detailed implementation of DRM on a blockchain platform	No security analysis and no specific comparison with other methods
Guo et al. [23]	2020	Combination of the public and private blockchains-enabled digital rights management system	Smart Contract, Blockchain	With a detailed functional module design and system architecture design	No security analysis, and comparison with other methods
Khan et al. [24]	2020	Content protection and transaction method using blockchain Ethereum technology	Ethereum, Smart Contract	There is a detailed implementation process An encryption algorithm is implemented for the content	No safety analysis, and comparison with other methods
Li et al. [25]	2021	Blockchain-watermarking scheme to protect the privacy	Watermarking, compressed sensing, BlockChain, IPFS	The low computational cost of the encryption process, simultaneous encryption, and compression, robustness of ciphertext	No detailed chain code implementation process
Heo et al. [26]	2021	Using SBBC's blockchain digital content trading system	Fingerprinting, BlockChain, SBBC, Consensus, algorithm.	Solves the problem of illegal digital content copying and leakage Addresses blockchain network overload and storage space limitations	Only part of the blockchain algorithm has been designed, and there are restrictions on the environment of use

Ma et al. [21] proposed a blockchain-based digital rights management scheme with efficient and secure authentication, privacy protection, and a conditional traceability method based on multiple signatures, so that the DRM (digital rights management) license, usage control, and constraint information can be easily retrieved from the blockchain. Ma et al. [22] proposed decentralized trust management and a secure usage control scheme for IoT big data based on a permission blockchain. Guo et al. [23] propose a novel network architecture for sharing and managing online educational multimedia resources with a combination of public and private blockchains, three specific smart contract schemes for implementing multimedia digital rights records, the secure storage of digital certificates, and unmediated authentication, respectively. Khan et al. [24] proposed a digital content protection and transaction method using blockchain Ethereum technology to prevent smart forgery and hacking. Li et al. [25] proposed a blockchain watermarking scheme to protect the privacy of compressed perceptual images, but it does not apply to content protection of short videos without a detailed chain code deployment process. Heo et al. [26] proposed a blockchain system using digital fingerprint recognition to improve DRM, which solves the problem of illegal digital content copying and leakage and also incorporates the SBBC (secret block-based blockchain) system to solve the problems of profit distribution, forgery, and counterfeiting. The approaches proposed by Ma et al. and Khan et al. both use Ethereum. However, only the blockchain part of the algorithm is designed, and there is

no system design according to the user in the application scenario of digital rights management. There are not enough nodes to support the security of public chains, and it is also difficult for public chains to achieve efficient application requirements. Tokens are not required to incentivize the nodes in enterprise-led digital rights management. The Consortium Blockchain has a faster transaction speed and lowers the consensus node requirements compared to public chains [27], so the Consortium Blockchain can better cope with digital copyright management in commercial digital copyright management.

As special digital copyright, digital trading cards will face the following problems: The first is how to provide an identification method to determine the authenticity of digital trading cards. Secondly, how to save the digital trading card information in a tamper-proof way so that users can track and verify the relevant information safely and transparently. Third, how to provide an efficient and reasonable management system for the whole digital trading card transaction, to avoid the emergence of centralized or third-party manipulation of digital trading card management, and to maintain the stability of the digital trading card market.

Based on the above-mentioned problems, our proposed digital trading card management system based on the consortium chain focuses on the security of digital trading cards, the traceability of information, and the unforgeable design and application of digital content, and it proposes the following objectives:

(1) Distribution and member access—Distributed computing storage with the same rights and obligations for each node. The identity of members who join the block is controlled to ensure information security.
(2) Integrity and non-tampering—The information data related to digital trading cards are signed by the ECDSA after the hash calculation before the transmission process, and the receiver verifies the signature and then stores it on the chain to solve the data security problem.
(3) Transparency—The source, issuance, purchase, and ownership information of digital trading cards are stored on the chain, and the information on the chain is open and transparent to enhance trust among users and protect collectors' rights. The proxy encryption ensures collectors' secure and transparent access to digital trading cards.
(4) Traceability—The history of the digital trading card information is traceable with time stamps and signature mechanisms, and all the information is non-repudiation.
(5) Timeliness—The sale and sharing of digital trading cards is managed reasonably, and digital content is not permanently appropriated by imposing time limits on access certificates.
(6) Resist attacks—Using timestamps, asymmetric encryption, and blockchain technology, our system can resist replay attacks, man-in-the-middle attacks, and witch attacks to build a secure digital trading card management platform.

The rest of this paper is organized as follows: Section 2 presents the preparatory knowledge involved in this system. Section 3 is a description of the digital trading card-related mechanisms and communication protocols. Section 4 provides a security analysis of the whole system. Section 5 gives the calculation of the system overhead and the comparison of other schemes. Finally, we conclude the article.

2. Preliminary

2.1. Hyperledger Fabric

Blockchain can be divided into three categories according to the participants: public chain, private chain, and consortium chain. The public chain is a blockchain network that is open to everyone, and anyone can participate in it. A private chain is a blockchain held by an organization. The consortium chain is a blockchain network that takes into account privacy, security, and decentralization and allows organizations and individuals to participate [28].

Hyperledger Fabric is a distributed ledger solution platform with a modular architecture and different pluggable components to adapt to different scenarios to achieve the architecture and functionality with high secrecy, resilience, flexibility, and scalability [29]. Hyperledger Fabric is a kind of consortium blockchain, and compared to the public blockchain represented

by Ethereum, Hyperledger Fabric focuses more on business applications, and only nodes involved in the corresponding business process can join the blockchain network [30]. This also means the nodes of Hyperledger Fabric do not need to reach consensus through PoW (Proof of Work), which increases the practicality of Hyperledger Fabric.

The workflow of Hyperledger Fabric is divided into the following steps. Step 1, proposal: Updating the ledger through the SDK requires sending a proposal to the endorsement node, which simulates the execution of the proposal based on the current version of the ledger. After the simulated execution, the read/write set is generated, and the ledger data are sent to change. Step 2, endorsement: The endorsement node signs the result of the book execution and returns this result to the SDK. After the SDK collects enough endorsement responses, it enters the third step for the update application. The SDK sends these signed endorsements to the orderer node, and the orderer node checks the signature endorsement according to the endorsement policy and sorts the updates. If the check meets the endorsement policy in the orderer node, it enters the fourth step-invoke update. The orderer node will send the read/write operation to each peer node, and each peer node will write the data into the block of each node after getting the invoke update. If there is an illegal endorsement request in the invoke update, the orderer node writes the illegal request into the blockchain and does not update the ledger to facilitate checking when the blockchain sends an error. After these four steps, that all the nodes' ledgers are updated and that the ledger contents are the same is ensured.

In this application scenario, the information shared between enterprises has a certain degree of privacy. Although the public chain realizes the information disclosure, it cannot protect the commercial secrets of each enterprise. Fabric can let enough people know the information to realize the mutual trust of enterprises and, at the same time, realize the protection of private information. Structurally, Hyperledger Fabric is superior to Ethereum in terms of the average transaction latency, throughput, privacy, and scalability, as well as the Hyperledger Fabric modularity and channel design, and it is more suitable for application scenarios between enterprises [31].

2.2. Smart Contract

The term "smart contract" was proposed in 1995 by Nick Szabo, a prolific cross-disciplinary legal scholar, who argued that a smart contract is a set of digitally defined commitments on which participants can execute agreements that can be continuously redesigned to increasingly approach the logic of the contract [32]. Unlike traditional contracts, which do not require execution by the participants, smart contracts can be triggered by a computer on the conditions met by different transactions. The information on the chain after the execution of a smart contract is tamper-proof and traceable

2.3. Elliptic Curve Digital Signature Algorithm (ECDSA)

The ECDSA signature algorithm is the elliptic curve digital signature algorithm [33]. This algorithm is used to sign digital messages in the blockchain, and it can be used to verify that the message has not been tampered with and that the signer cannot sign the message to disclaim it.

The ECDSA signature process is as follows: the sender first selects a random value from [1, n − 1], takes the hash value of the message to be sent, and calculates $z = h(M)$, $(x, y) = kG$, $r = x \bmod n$, $s = k^{-1}(z + r * dA) \bmod n$. The (r, s) obtained using the ECDSA signature and the original message M without hash are then sent to the message receiver (h is the hash function, dA is the sender's private key).

The ECDSA verifies the signature as follows: after receiving the signature pair (r, s) and message M, the message receiver will use the sender's public key QA to verify the sender's signature: $z' = h(M)$, $u_1 = z' * s^{-1} \bmod n$,$u = r * s^{-1} \bmod n$, $(x', y') = u_1 G + u_2 Q$, by determining whether the values of r and $x' \bmod n$ are equal. If they are, the message receiver can confirm that the message sent by the sender has not been tampered with and is guaranteed by the sender.

2.4. Proxy Re-Encryption

Blaze et al. [34] first proposed atomic agent cryptography in 1998, where a semi-trusted agent uses functional computation to convert the ciphertext encrypted by the authorized person, Alice, using the key to the ciphertext encrypted by the authorized person Bob's key. Thus, without exposing Alice's key, Bob can decrypt Alice's ciphertext, and the agent cannot view the underlying plaintext.

(a) Definition of system parameters: Let a large prime number q and multiplicative group $Z_P{}^*$ generate g, where q and g are public parameters.
(b) Key generation: Licensee Alice chooses positive integers $ga \bmod p$ $(a < p)$, $gb \bmod p$ $(b < p)$ at random as his keys. Then, the decryption key b is sent to the licensee Bob via a secure transmission channel.
(c) Authorized person encrypts plaintext: Set random number k and calculate $C_1 = mg^k \bmod p$ and $C_2 = g^{ak} \bmod p$ to obtain ciphertext (C_1, C_2).
(d) Re-encryption key generation: If Bob is authorized by Alice's message, then Alice needs to send ciphertext (C_1, C_2) and proxy key b/a to the proxy.
(e) Re-encryption process: After receiving (C_1, C_2) and proxy key b/a, the agent re-encrypts the ciphertext to $(C_1{}', C_2{}') = (mg^k \bmod p, mg^{ak(b/a)} \bmod p) = (mg^k \bmod p, mg^{bk} \bmod p)$ and sends the re-encrypted ciphertext $(C_1{}', C_2{}')$ to Bob.
(f) Licensee decrypts the ciphertext: Licensee Bob computes $C_2{}'^{(1/b)} = g^k \bmod p$ using the decryption key b and obtains the plaintext using $C_1{}'/C_2{}'^{(1/b)} = mg^k \bmod p / g^k \bmod p = m$.

3. Proposed Scheme

3.1. System Architecture

In this study, we use the ECDSA, blockchain, smart contracts, and proxy re-encryption to design a tamper-proof and traceable authorization mechanism for NBA digital trading cards. Figure 1 shows the system architecture diagram, in which the system architecture roles include: blockchain center (BCC), NBA digital trading card center (N), digital trading card manufacturer (M), digital trading card proxy (P), user (U), and bank (B).

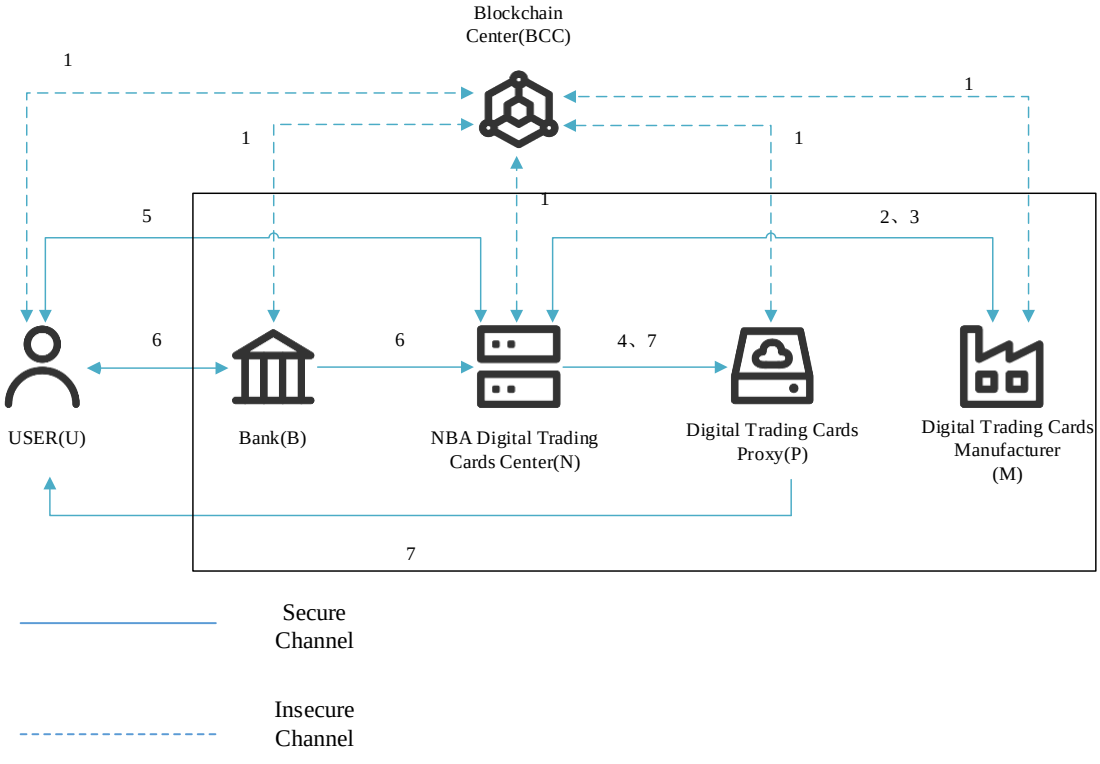

Figure 1. System architecture diagram.

(1) Blockchain Center (BCC): The Blockchain Center registers each role in the system architecture and issues identity certificates as well as public and private keys to each role.
(2) NBA Digital Trading Cards Center (N): N is the NBA's cloud platform, responsible for the provision of video information and authorization of video manufacturing, review and issuance of digital trading cards, and review of user requests to determine whether to allow access to digital content resources.
(3) Digital Trading Cards Manufacturer (M): A digital trading card manufacturer must obtain authorization from the NBA Center, as well as genuine video, and manufacture digital content related to digital trading cards.
(4) Digital Trading Cards Proxy (P): The Digital Trading Cards Proxy is the NBA's agent, which is the cloud platform for storing digital trading cards. After the user's identity is verified, the proxy is actually responsible for authorizing the user in the cloud (sending re-encrypted ciphertext).
(5) User (U): Users are collectors of digital trading cards. When a user wants to access the NBA's digital trading cards, the user should pay a fee to the NBA.
(6) Bank (B): The bank issues a payment certificate to the user.

We briefly explain the step-by-step transaction scenario for digital trading cards.

Step 1 Registration Phase—This Step is the registration phase for each role in the system; all users, digital trading card manufacturers, NBA digital trading card centers, digital trading card agents, and banks need to register with the blockchain center to obtain the public and private keys provided by the blockchain center.
Step 2 Manufacturing Authorization Phase—When a digital trading card manufacturer wants to manufacture a digital trading card for sale, they need to obtain an authorization code from the NBA digital trading card center for the manufacture and an authentic NBA game video.
Step 3 Review Phase—After the manufacturer has obtained the authorization from the NBA Center, it will enter the review phase after completing the authorization phase. The digital trading card manufacturer will send the processed digital content and information related to the digital content to the NBA digital trading card center so that the NBA digital trading card center conducts a review of the content.
Step 4 Issue Phase—After the NBA digital trading card center completes the review of the digital trading card, the video content is classified, and the number of sales is approved. The NBA digital trading card center uploads the classification and number of sales of the digital trading card to the blockchain center and sends the content encrypted (by key encryption) to the digital trading card agent for storage.
Step 5 Identity Verification and Invoicing Phase—When a user submits a transaction request to the NBA digital trading card center, the NBA digital trading card center generates an invoice and transaction ID for the user after reviewing the user's identity.
Step 6 Payment Phase—The user makes the payment through the bank. After the payment, the user requests the bank to issue a payment certificate for the transaction to authenticate the payment.
Step 7 Browse and Access phase—The NBA digital trading card center verifies the user's identity and payment certificate, and upon successful verification, the NBA digital trading card center transfers the re-encryption key to the proxy and the time-limited decryption key to the user. The proxy's re-encryption key generates a re-agent ciphertext, which the proxy transfers to the user, who uses the key to automatically decrypt the protected digital trading card and browse the digital trading card.

3.2. Smart Contract Initialization

Blockchain technology is used in the proposed architecture. We use the blockchain to verify and save key information. Algorithm 1 is the definition of the smart contract structure used in the digital trading card management system architecture.

Algorithm 1. The structure of the smart contract scheme

```
struct smart contract mninf/uninf{
string mn/un id;
string mn/un detail;
string mn/un cert;
 string mn/un tsp;
}
struct smart contract nminf/repuinf{
string nm/repu id;
string nm/repu detail;
string nm/repu ac;
string nm/repu tsp;
}
struct smart contract reuninf{
  string reun id;
  string reun detail;
  string reun payment;
  string reun tsp;
  }
struct smart contract buinf{
string bu id;
string bu detail;
string bu payment;
string bu tsp;
}
struct smart contract remninf/npinf{
string remn/np id;
string remn/np detail;
string remn/np skey;
string remn/np tsp;
}
struct smart contract nuinf{
string nu id;
string nu detail;
string nu tid;
string nu tsp;
}
 struct smart contract renpinf{
 string renp id;
 string renp detail;
 string renp tsp;
}
 struct smart contract renuinf{
 string renu id;
 string renu detail;
 string renu tsp;
}
```

3.3. Registration Phase

The system role X can represent the digital trading card manufacturer, digital trading card proxy, NBA digital trading card center, bank, and user. Figure 2 shows the flow chart of the registration phase.

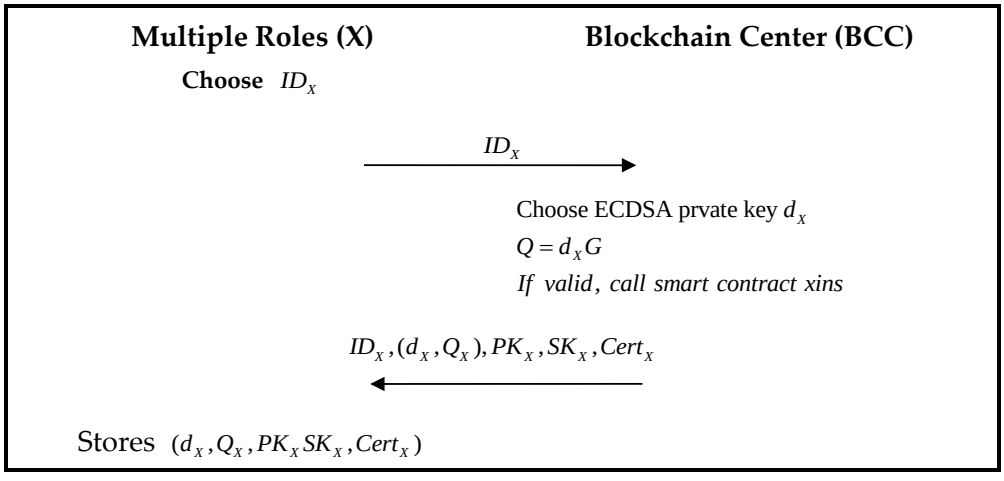

Figure 2. The flow chart of the registration phase.

Step 1: Role X generates an identity ID_X and sends it to the blockchain center.

Step 2: The blockchain center generates ECDSA private key d_X based on role ID_X and passes the calculation:

$$Q_X = d_X G \tag{1}$$

Obtain the public key Q_X, and bind ID_X to the generated public and private keys in mapping.

If the role's X's identity is verified, the smart contract Xins will be triggered, the content of which is Algorithm 2.

Algorithm 2. The scheme of the smart contract xins.

```
function insert x smart constract xins (
string x_id, string x_detail){
    count + +;
    x[count].id = id;
    x[count].detail = detail;
}
String x_keypairs;
```

And sends the generated $ID_X, (d_X, Q_X), PK_X SK_X, Cert_X$ to role X.

Step 3: Role X saves the $(d_X, Q_X, PK_X SK_X, Cert_X)$ returned by the blockchain center.

3.4. Manufacturing Authorization Phase

The NBA digital trading card center is not responsible for the production of digital trading cards but rather licenses the production of digital trading cards to digital trading card manufacturers. Because the content of digital trading cards is highlights highlighted by the NBA league, it is mostly short-form video manufacturers who want to obtain the license. Short-form video manufacturers need to apply to the NBA digital trading card center to obtain the license code for production and the NBA video that can be edited. The flow chart of the licensing stage of digital trading card production is shown in Figures 3 and 4.

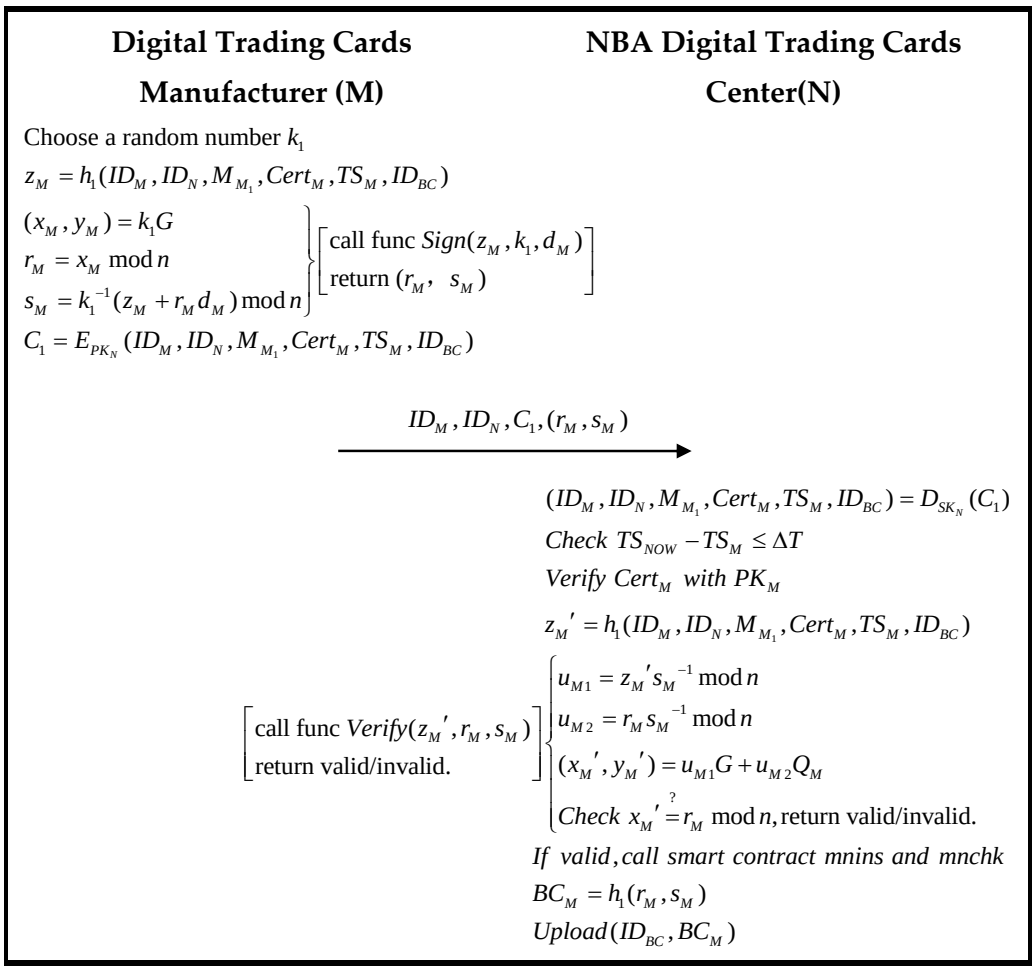

Figure 3. The flowchart of the Manufacturing Authentication phase (M to N).

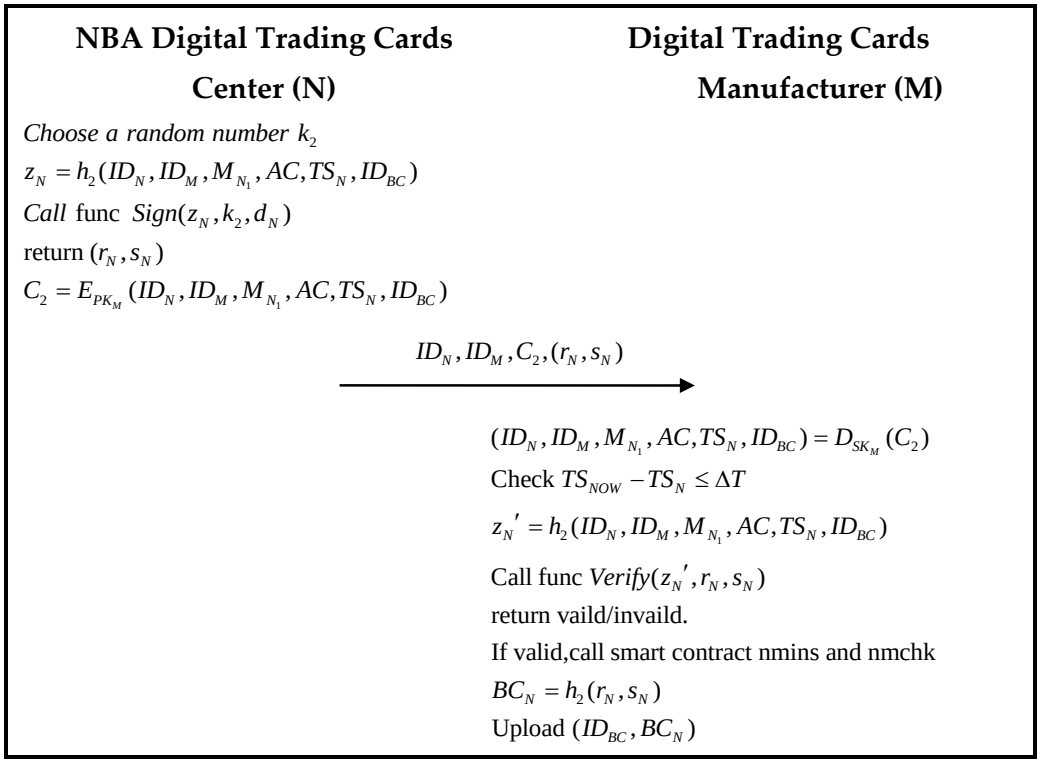

Figure 4. The flowchart of the Manufacturing Authentication phase (N to M).

Step 1: The digital trading card maker generates a random value k_1 and computes:

$$z_M = h_1(ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}), \tag{2}$$

$$(x_M, y_M) = k_1 G, \tag{3}$$

$$r_M = x_M \bmod n, \tag{4}$$

$$s_M = {k_1}^{-1}(z_M + r_M d_M) \bmod n, \tag{5}$$

$$C_1 = E_{PK_N}(ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}), \tag{6}$$

and sends ID_M, ID_N, C_1, (r_M, s_M) to the NBA digital trading card center, where M_{M_1} is a request from Role M (digital trading card manufacturer) to make a digital trading card at the NBA digital trading card center.

(r_M, s_M). The digital trading card manufacturer uses its private key to sign and generate and encrypt the sent message, as with the receiver's public key, to ensure message security and complete traceability. Digital trading card manufacturers provide their ID_M to the NBA digital trading card center to clarify their identity.

Step 2: The NBA digital trading card center first calculates:

$$(ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}) = D_{SK_N}(C_1), \tag{7}$$

and by calculating:

$$TS_{NOW} - TS_M \leq \Delta T, \tag{8}$$

To verify the validity of the timestamp, use PK_M to verify $Cert_M$, determine the correctness of the ECDSA signature, and then calculate:

$$z_M{}' = h_1(ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}), \tag{9}$$

$$u_{M1} = z_M{}' {s_M}^{-1} \bmod n, \tag{10}$$

$$u_{M2} = r_M {s_M}^{-1} \bmod n, \tag{11}$$

$$({x_M}', {y_M}') = u_{M1}G + u_{M2}Q_M, \tag{12}$$

$${x_M}' \stackrel{?}{=} r_M \bmod n, \tag{13}$$

If the verification is passed, N will obtain the digital trading card manufacturing application information from M and trigger the smart contract mnins and mnchk. The content is as follows (Algorithm 3):

Algorithm 3. The scheme of the smart contracts mnins and mnchk

```
function insert smart constract mnins(
string mn_id, string mn_detail,
string mn_cert, string mn_tsp) {
count++;
mn[count].id = id
mn[count].detail = detail;
mn[count].cert = cert;
mn[count].tsp = tsp;
}
sign string m_key(mn_id, mn_detail,
mn_cert, mn_tsp);
```

```
function check smart cos tract mnchk(
 string mn_id, string mn_detail
string mn_cert, string mn_tsp}{
return mn_id.exist;
return mn_detail.exist;
return mn_cert.exist;
return mn_tsp.exist;
}
verify string n_key(mn_id, mn_detail,
mn_cert, mn_tsp);
```

N uploads the resulting (ID_{BC}, BC_M) to the blockchain center by computing:

$$BC_M = h_1(r_M, s_M), \tag{14}$$

Then, N generates a random value k_2 and calculates:

$$z_N = h_2(ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC}), \tag{15}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature:

$$(r_N, s_N) = Sign(z_N, k_2, d_N), \tag{16}$$

$$C_2 = E_{PK_M}(ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC}), \tag{17}$$

Send ID_N, ID_M, C_2, (r_N, s_N) to the digital trading card manufacturer, where M_{N_1} is the reply of role N to the request of M.

Algorithm 4. Chaincode Sign and Verify scheme

```
func Sign (h string, k string, d string){
(x, y)= k * G
r = x % n
s =(h + r * d)/x % n
return r, s
 }
```

```
func Verify (h string, r string, s string)
(result string) {
u1 = h/s % n
u2 = r/s % n
(x, y)= u1 * G + u2 * Q
if x == r
   return "valid"
else
   return "invalid"
```

Step 3: The digital trading card manufacturer first calculates:

$$(ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC}) = D_{SK_M}(C_2), \tag{18}$$

and then computes:

$$TS_{NOW} - TS_N \leq \Delta T, \tag{19}$$

M to verify the validity of the timestamp, and determine the correctness of the ECDSA signature:

$$z_N{}' = h_2(ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC}), \tag{20}$$

Use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_N{}', r_N, s_N). \tag{21}$$

If the verification is passed, the content request message is verified by N, and the smart contracts nmins and nmchk will be sent as follows (Algorithm 5):

Algorithm 5. The scheme of the smart contracts nmins and nmchk

```
function insert smart constract nmins(
string nm_id, string nm_detail,
string nm_ac, string nm_tsp) {
    count + +;
    nm[count].id = id;
    nm[count].detail = detail;
    nm[count].ac = ac;
    nm[count].tsp = tsp
}
sign string n_key(nm_id, nm_detail,
nm_ac, nm_tsp);
```

```
function check smart cos tract nmchk(
string nm_id, string nm _detail
string nm_ac, string nm_tsp}{
    return nm _id.exist;
    return nm _detail.exist;
    return nm _ac.exist;
    return nm _tsp.exist;
}
verify string m_key(nm_id, nm_detail,
nm_ac, nm_tsp)
```

M computes:

$$BC_N = h_2(r_N, s_N), \tag{22}$$

and then uploads the calculated (ID_{BC}, BC_N) to the blockchain center.

3.5. Review Phase

The manufacturer will encrypt the digital trading card with a symmetric key and send it to the NBA digital trading card center after the manufacturing of the digital trading card. The NBA digital transaction card center decrypts the digital envelope to obtain the symmetric key to obtain the specific contents of the digital transaction card. Review video content to ensure compliance. A flow chart of the digital trading card review phase is shown in Figure 5.

Step 1: The manufacturer of the digital trading card first encrypts the digital content using a symmetric key:

$$key_m = (KeyID, Seed), \tag{23}$$

$$C_{sym} = E_{key_m}(M), \tag{24}$$

Then, a random value k_3 is generated and calculated:

$$z_M = h_3(ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC}), \tag{25}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature:

$$(r_M, s_M) = Sign(z_M, k_3, d_M), \tag{26}$$

$$C_3 = E_{PK_N}(ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC}), \tag{27}$$

Send ID_M, ID_N, C_3, S, (r_M, s_M) to the NBA digital trading card center, where M_{M_2} the information related to the digital trading card manufactured by role M.

Then, a random value k_3 is generated and calculated:

Digital trading card manufacturers use digital envelopes to encrypt digital content, ensuring the security of digital information.

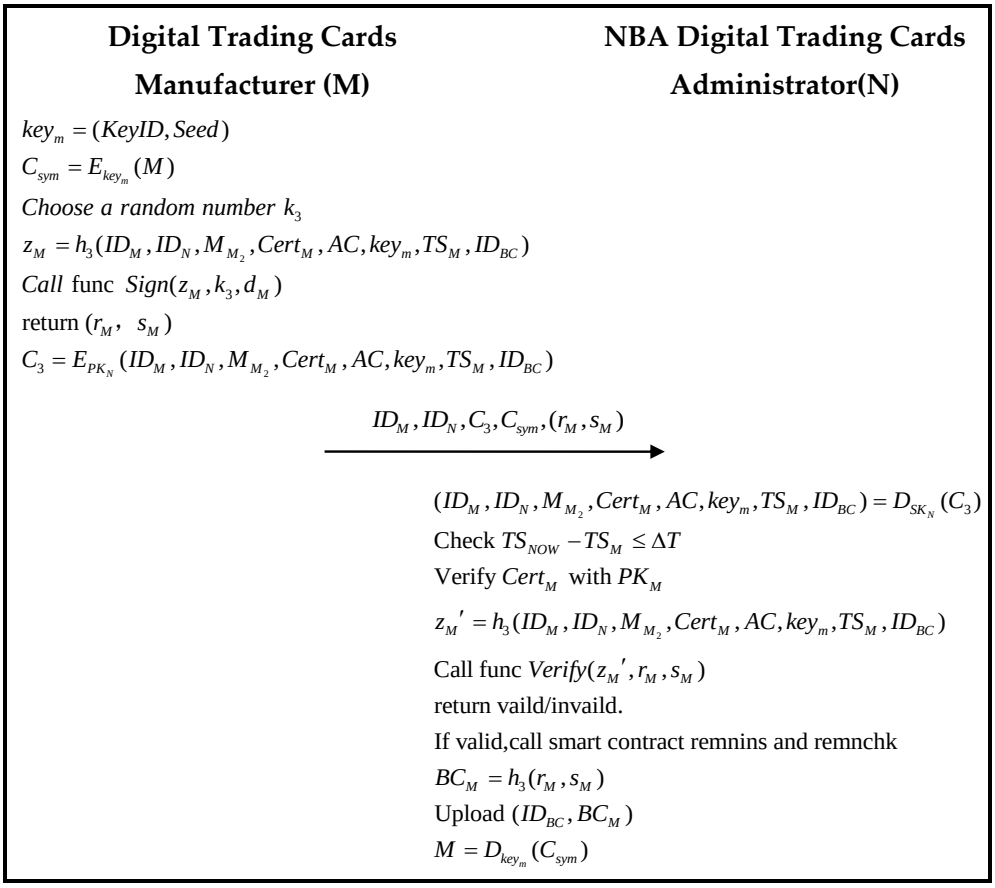

Figure 5. The flowchart of the Review Phase.

Step 2: NBA digital trading card center calculates first:

$$(ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC}) = D_{SK_N}(C_3), \tag{28}$$

Verify the timestamp:

$$TS_{NOW} - TS_M \leq \Delta T, \tag{29}$$

and then calculate:

$$z_M{}' = h_3(ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC}), \tag{30}$$

Use the "Verify" function in Algorithm 4 to verify the ECDSA signature.

$$Verify(z_M{}', r_M, s_M). \tag{31}$$

If the verification is passed, N will obtain the digital trading card manufacturing information of M and trigger the smart contracts reanins remnins and reamnchk, as follows (Algorithm 6).

Algorithm 6. The scheme of the smart contracts remnins and remnchk

```
function insert smart constract remnins(
string remn_id, string remn_detail,
string remn_skey, string remn_tsp) {
    remn[count].id = id;
    remn[count].detail = detail;
    remn[count].akey = skey;
    remn[count].tsp = tsp;
}
sign string m_key(remn_id, remn_detail,
remn_skey, remn_tsp);
verify string n_key(remn_id, remn_detail,
remn_skey, remn_tsp);
```

```
function check smart cos tract remnchk(
string remn_id, string remn_detail
string remn_skey, string remn_tsp}{
    return remn_id.exist;
    return remn_detail.exist;
    return remn_skey.exist;
    return remn_tsp.exist;
}
```

N computes:

$$BC_M = h_3(r_M, s_M), \tag{32}$$

and then uploads the calculated (ID_{BC}, BC_M) to the blockchain center.

The digital content is then decrypted using key_m:

$$M = D_{key_m}(C_{sym}). \tag{33}$$

The NBA digital trading card center reviews the digital content by combining the decrypted digital content with verified manufacturing information.

3.6. Issue Phase

The NBA digital trading card center reviews the digital content and categorizes it, rating the quality of the video content to determine the quantity and price of distribution. The digital content is determined to be sold as a digital trading card, and the NBA digital trading card center encrypts the digital trading card $KeyID, Seed$. Afterward, the encrypted digital trading cards are sent to a digital trading card proxy for cloud storage. Figure 6, below, illustrates the flow chart for approving the distribution of digital trading cards.

Step 1: The NBA digital trading card center first encrypts the digital content using a symmetric key:

$$key_m = (KeyID, Seed), \tag{34}$$

$$c_1 = ID_{DC} g^{ak} \bmod p, \tag{35}$$

$$c_2 = g^k \bmod p, \tag{36}$$

Then, a random value k_4 is generated and calculated:

$$z_N = h_4(ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC}), \tag{37}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature:

$$(r_N, s_N) = Sign(z_N, k_4, d_N), \tag{38}$$

$$C_4 = E_{PK_P}(ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC}). \tag{39}$$

Send ID_N, ID_P, C_4, S, (r_N, s_N) to the digital trading card proxy, where M_{N_2} is the role N (NBA digital trading card center) to determine the number and price of digital trading cards to be issued. c_1, c_2 is the proxy's re-encrypted cipher text.

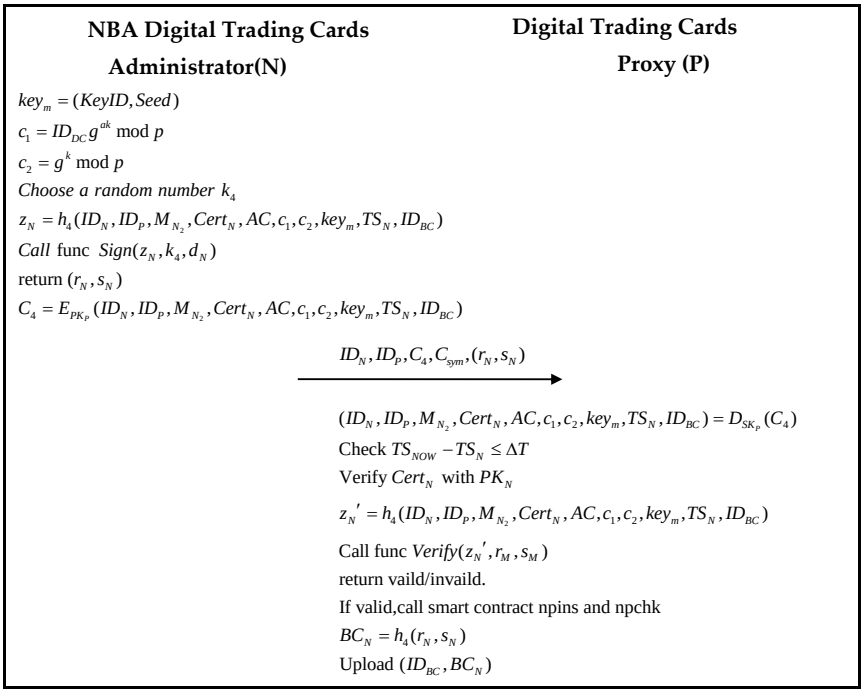

Figure 6. The flowchart of the Issue Phase.

Step 2: NBA digital trading card center decrypts as follows.

$$(ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC}) = D_{SK_P}(C_4), \tag{40}$$

and verify:

$$TS_{NOW} - TS_N \leq \Delta T, \tag{41}$$

And use PK_N to verify $Cert_N$ to determine if the correctness of the ECDSA signature as follows

$$z_N{}' = h_4(ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC}), \tag{42}$$

Then, use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_N{}', r_N, s_N), \tag{43}$$

If the verification is passed, the digital trading card proxy will obtain the issuance information of N's digital trading card and trigger the smart contracts npins and npchk, which are as follows (Algorithm 7).

Algorithm 7. The scheme of the smart contracts npins and npchk

```
function insert smart constract npins(
 string np_id, string np _detail
 string np_skey, string np_tsp) {
count + +;
    np[count].id = id;
    np[count].detail = detail;
    np[count].skey = skey;
    np[count].tsp = tsp;
}
sign string n_key(np_id, np_detail,
 np_skey, np_tsp);
verify string p_key(np_id, np_detail,
 np_skey, np_tsp);
```

```
function check smart cos tract npchk(
 string np_id, string np _detail
 string np_skey, string np_tsp}{
    return np _id.exist;
    return np _detail.exist;
    return np _skey.exist;
    return np _tsp.exist;
}
```

P uploads the resulting (ID_{BC}, BC_N) to the blockchain center by computing:

$$BC_N = h_4(r_N, s_N). \tag{44}$$

3.7. Identity Verification and Invoicing Phase

The user submits a purchase request to the NBA digital trading card center through the APP. After reviewing the identity of the application, an invoice is issued. Figure 7 below shows the flowchart of a user submitting a purchase request.

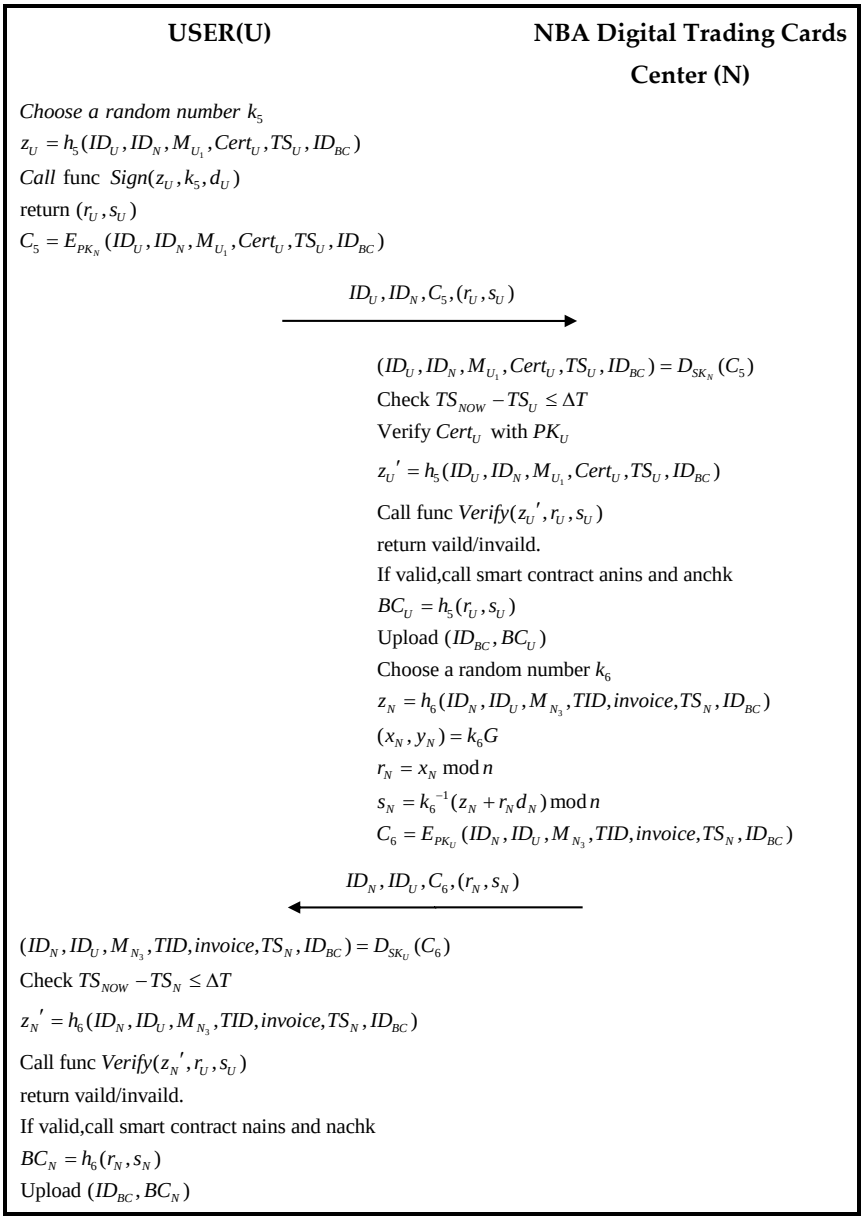

Figure 7. The flowchart of the Identity Verification and Invoicing Phase.

Step 1: The user generates random values k_5, and calculates:

$$z_U = h_5(ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC}), \tag{45}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature:

$$(r_U, s_U) = Sign(z_U, k_5, d_U), \tag{46}$$

$$C_5 = E_{PK_N}(ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC}), \tag{47}$$

and sends ID_U, ID_N, C_5, (r_U, s_U) to the NBA digital trading card center, where M_{U_1} is the role U submitting a purchase request to the NBA digital trading card center. C_5 is encrypted for integrity checking purposes, and ID_U is the identity provided by the user to the digital transaction card center.

Step 2: NBA digital trading card center first calculates:

$$(ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC}) = D_{SK_N}(C_5), \tag{48}$$

Then, calculate:

$$TS_{NOW} - TS_U \leq \Delta T, \tag{49}$$

to verify the validity of the timestamp. Use PK_U to verify $Cert_U$, check the correctness of the ECDSA signature, and then calculate:

$$z_U{}' = h_5(ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC}), \tag{50}$$

Then, use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_U{}', r_U, s_U). \tag{51}$$

If the verification is passed, N will obtain the digital trading card purchase information of U and trigger the smart contract anins and anchk. Algorithm 8 is as follows.

Algorithm 8. The scheme of the smart contracts unins and unchk

```
function insert smart constract unins(
string un_id, string un_detail,
string un_cert, string un_tsp) {
    count++
    un[count].id = id;
    un[count].detail = detail;
    un[count].cert = cert;
    un[count].tsp = tsp;
}
sign string a_key(un_id, un_detail,
un_cert, un_tsp);
verify string n_key(un_id, un_detail,
un_cert, un_tsp);
```

```
function check smart cos tract unchk(
string un_id, string un_detail
string un_cert, string un_tsp}{
    return un_id.exist;
    return un_detail.exist;
    return un_cert.exist;
    return un_tsp.exist;
}
```

Step 3: N computes:

$$BC_U = h_5(r_U, s_U), \tag{52}$$

and then uploads the results $BC_U = h(r_U, s_U)$ to the blockchain center.

NBA digital trading card center generates random values k_6 and calculates:

$$z_N = h_6(ID_N, ID_U, M_{N_3}, TID, invoice, TS_N, ID_{BC}), \tag{53}$$

and executes the function "Sign" as shown in Algorithm 4 to generate the signature:

$$(r_N, s_N) = Sign(z_N, k_6, d_N), \tag{54}$$

$$C_6 = E_{PK_U}(ID_N, ID_U, M_{N_3}, TID, invoice, TS_N, ID_{BC}), \tag{55}$$

ID_N, C_6, (r_N, s_N) is then sent to the user, where M_{N_3} is a reply message from N to the U purchase access message. TID is the transaction ID, and *invoice* is the invoice information.

If the verification passes, the user will receive the transaction ID and invoice. nulin and nuchk, smart contracts, are triggered as follows (Algorithm 9).

Algorithm 9. The scheme of the smart contracts nuins and nuchk

```
function insert smart constrtidt nuins(
string nu_id, string nu_detail,
string nu_tid, string nu_tsp) {
    count + +;
    nu[count].id = id;
    nu[count].detail = detail;
    nu[count].tid = tid;
    nu[count].tsp = tsp;
}
sign string n_key(nu_id, nu_detail,
nu_tid, nu_tsp);
```

```
function check smart cos trtidt nuchk(
string nu_id, string nu _detail
string nu_tid, string nu_tsp}{
    return nu _id.exist;
    return nu _detail.exist;
    return nu _tid.exist;
    return nu _tsp.exist;
}
verify string a_key(nu_id, nu_detail,
nu_tid, nu_tsp)
```

The user calculates:

$$BC_N = h_6(r_N, s_N), \tag{56}$$

and then uploads the calculated (ID_{BC}, BC_N) to the blockchain center.

3.8. Payment Phase

Figure 8 below shows the process of issuing a payment certificate by the bank after the user has made a payment.

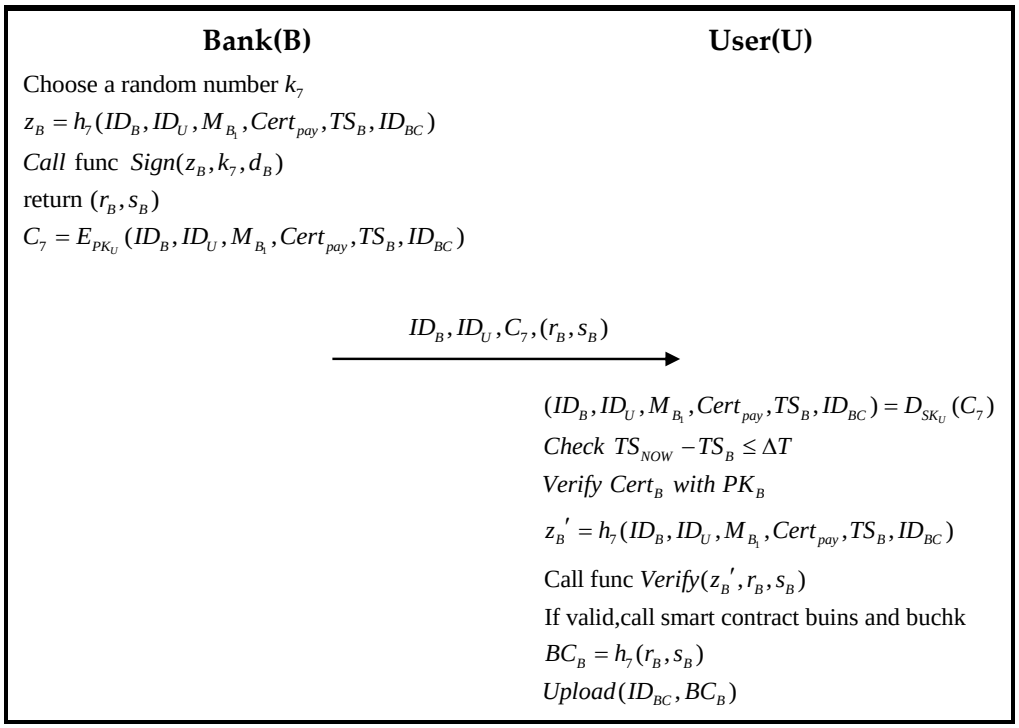

Figure 8. The flowchart of the Payment Phase.

Step 1: The bank generates random values k_7, and calculates:

$$z_B = h_7(ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC}), \tag{57}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature

$$(r_B, s_B) = Sign(z_B, k_7, d_B), \tag{58}$$

$$C_7 = E_{PK_U}(ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC}). \tag{59}$$

And sends ID_B, ID_U, C_7, (r_M, s_M) to the user. Where M_{B_1} is the information related to the issuance of the certificate by the bank. C_7 is encrypted for transmission security. ID_B is the identity provided by the bank to the user.

Step 2: The user first calculates:

$$(ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC}) = D_{SK_U}(C_7), \tag{60}$$

and verify:

$$TS_{NOW} - TS_B \leq \Delta T, \tag{61}$$

Use PK_B to verify $Cert_B$, check the correctness of the ECDSA signature, and then calculate:

$$z_B{}' = h_7(ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC}), \tag{62}$$

Then, use the "Verify" function in Algorithm 4 to verify the signature.

$$Verify(z_B{}', r_B, s_B), \tag{63}$$

If the authentication is passed, the user will obtain the bank's payment credentials for the transaction and trigger the smart contract buins and buchk, as follows (Algorithm 10).

Algorithm 10. The scheme of the smart contracts buins and buchk

```
function insert smart constract buins(
string bu_id, string bu_detail,
string bu_payment, string bu_tsp) {
    count++;
    bu[count].id = id;
    bu[count].detail = detail;
    bu[count].payment = payment;
    bu[count].tsp = tsp;
}
sign string a_key(bu_id, bu_detail,
bu_payment, bu_tsp);
```

```
function check smart cos tract buchk(
string bu_id, string bu _detail
string bu_payment, string bu_tsp}{
    return bu _id.exist;
    return bu _detail.exist;
    return bu _payment.exist;
    return bu _tsp.exist;
 }
verify string n_key(bu_id, bu_detail,
bu_payment, bu_tsp);
```

N computes:

$$BC_B = h_7(r_B, s_B), \tag{64}$$

and then uploads the results $BC_B = h(r_B, s_B)$ to the blockchain center.

3.9. Browse and Access Phase

After the bank issues a payment certificate for the transaction, the user can use the payment certificate to initiate a browse request to the NBA digital trading card center. After the NBA digital trading card center verifies the transaction and the payment certificate, it sends a browse access message to the NBA digital trading card proxy. The NBA digital trading card proxy then issues a license key to the user, who uses it to decrypt the protected digital trading card. Figures 9–11 show the flow chart of the payment verification and browse access phases.

Step 1: The user generates a random value k_8 and calculates:

$$z_U = h_8(ID_U, ID_N, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC}), \tag{65}$$

and executes the function "Sign" as shown in Algorithm 4 to generate the signature:

$$(r_U, s_U) = Sign(z_U, k_8, d_U), \tag{66}$$

$$C_8 = E_{PK_N}(ID_U, ID_N, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC}), \tag{67}$$

and sends ID_U, ID_N, C_8, (r_U, s_U) to the NBA digital trading card center, where M_{U_2} is a browse request from U to N.

Step 2: NBA digital trading card center first calculates:

$$(ID_U, ID_N, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC}) = D_{SK_N}(C_8), \tag{68}$$

and verify the validity of the timestamp by calculating:

$$TS_{NOW} - TS_U \leq \Delta T. \tag{69}$$

Use PK_U to verify $Cert_U$, check the correctness of the ECDSA signature, and then calculate:

$$z_U{}' = h_8(ID_U, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC}), \tag{70}$$

and use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_U{}', r_U, s_U), \tag{71}$$

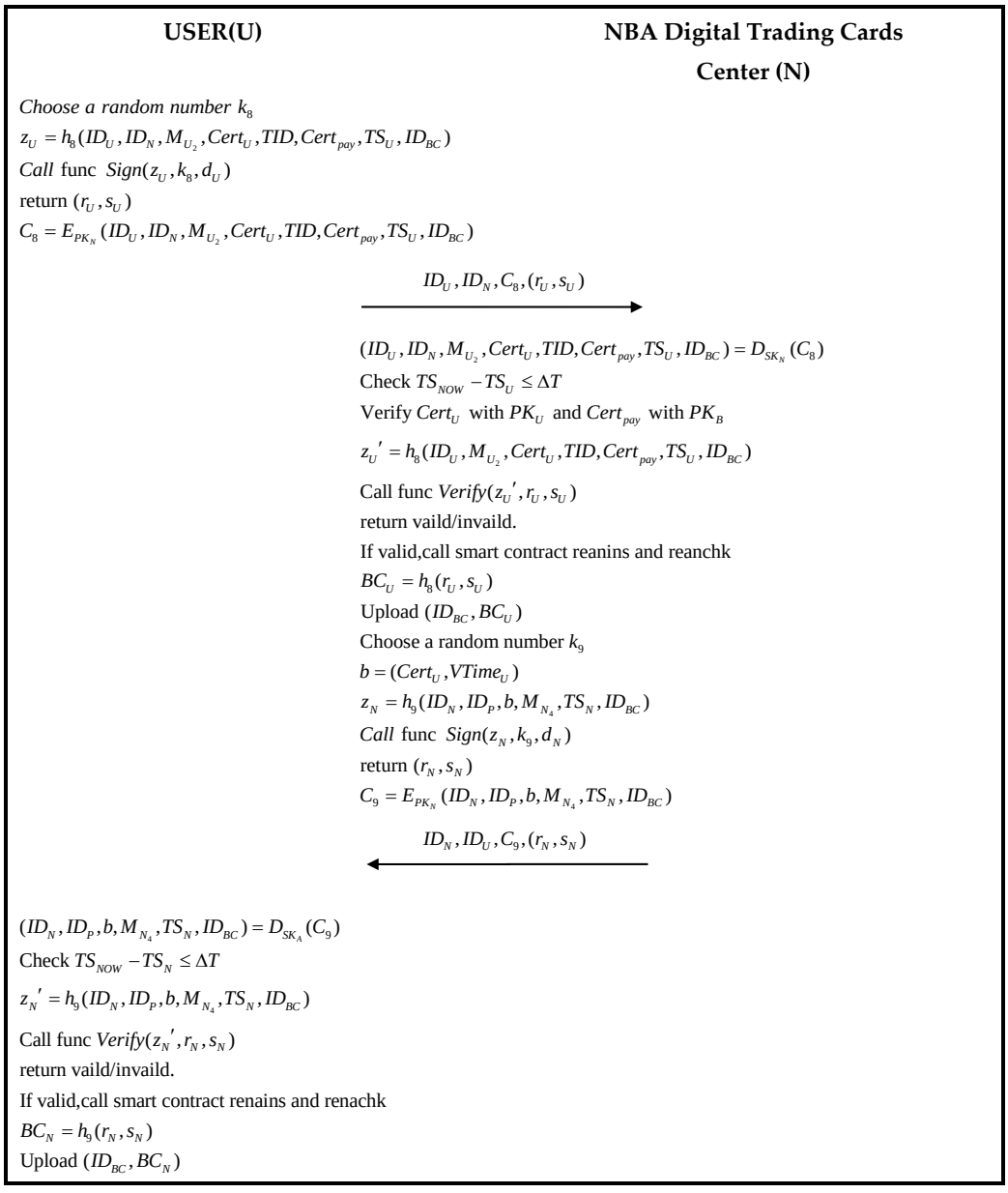

Figure 9. The flowchart of the Browse and Access phase (U and N).

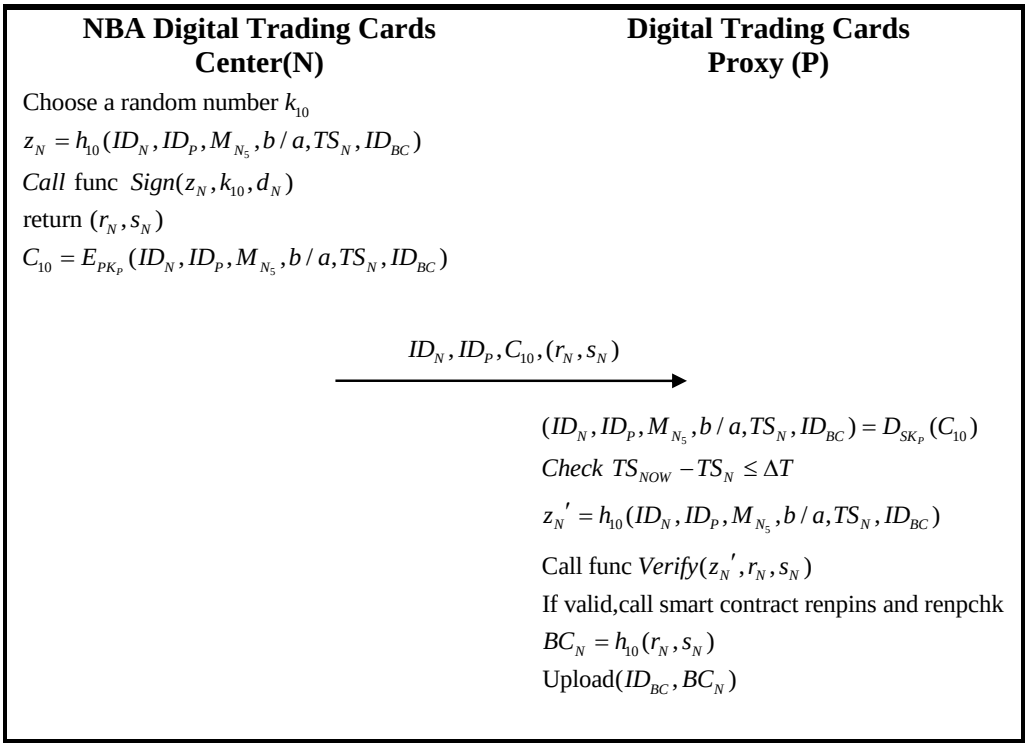

Figure 10. The flowchart of the Browse and Access Phase (N to P).

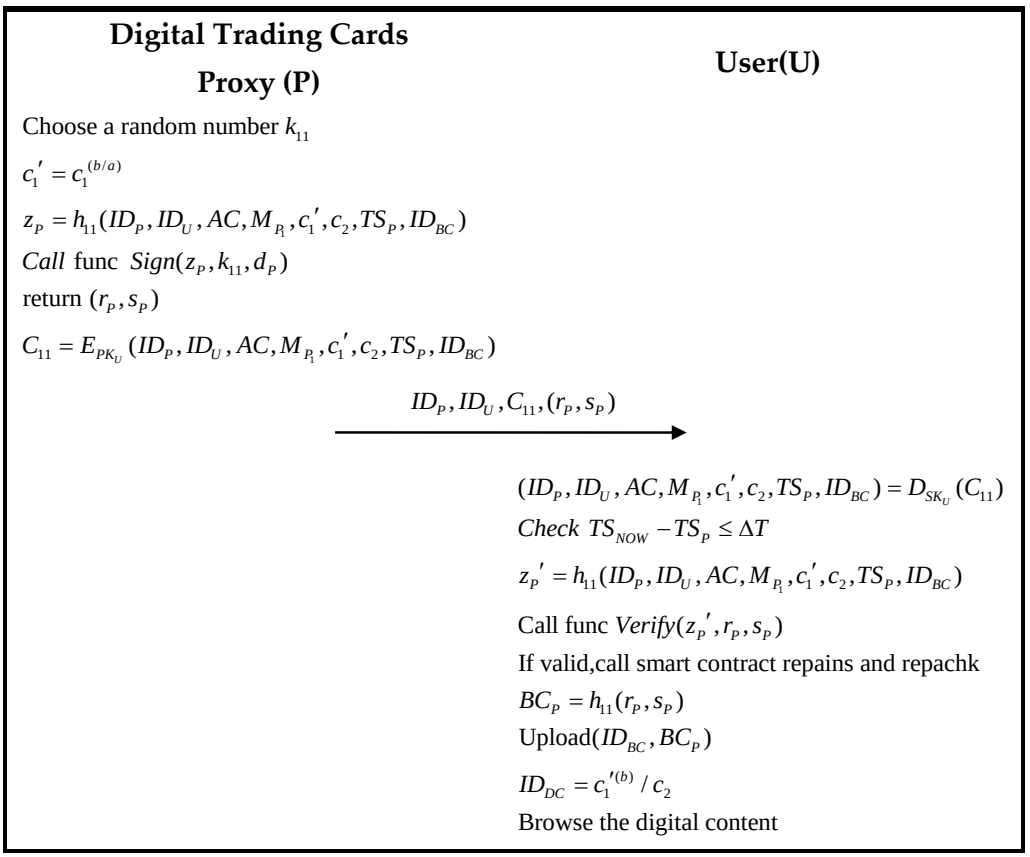

Figure 11. The flowchart of the Browse and Access Phase (P to U).

If the verification is passed, N will obtain the information of A's digital trading card browsing application and trigger the smart contract reunins and reunchk; Algorithm 11 is as follows.

Algorithm 11. The scheme of the smart contracts reunins and reunchk

```
function insert smart constract reunins(
string reun_id, string reun_detail,
string reun_payment, string reun_tsp) {
    count + +;
    reun[count].id = id;
    reun[count].detail = detail;
    reun[count].payment = payment;
    reun[count].tsp = tsp;
}
sign string a_key(reun_id, reun_detail,
reun_payment, reun_tsp);
```

```
function check smart cos tract reunchk(
string reun_id, string reun_detail,
string reun_payment, string reun_tsp){
    return reun _id.exist;
    return reun _detail.exist;
    return reun _payment.exist;
    return reun _tsp.exist;
}
 verify string n_key(reun_id, reun_detail,
 reun_payment, reun_tsp);
```

N computes:

$$BC_U = h_8(r_U, s_U), \tag{72}$$

and then uploads the results $BC_U = h_8(r_U, s_U)$ to the blockchain center.

Step 3: N generates a random value k_9, and calculates:

$$b = (Cert_U, VTime_U), \tag{73}$$

$$z_N = h_9(ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC}), \tag{74}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature:

$$(r_N, s_N) = Sign(z_N, k_9, d_N), \tag{75}$$

$$C_9 = E_{PK_N}(ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC}). \tag{76}$$

Sends $ID_N, ID_U, C_9, (r_N, s_N)$ to the user, where M_{N_4} is a reply from N to the U access message.

Step 4: The user first:

$$(ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC}) = D_{SK_A}(C_9), \tag{77}$$

and calculate:

$$TS_{NOW} - TS_N \leq \Delta T. \tag{78}$$

To verify the validity of the timestamp, determine the correctness of the ECDSA signature, and then calculate:

$$z_N{}' = h_9(ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC}), \tag{79}$$

then use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_N{}', r_N, s_N). \tag{80}$$

If the verification passes, the smart contracts renuins and renuchk are triggered, and Algorithm 12 is as follows:

Algorithm 12. The scheme of the smart contracts renuins and renuchk

```
function insert smart constract renuins(
string renu_id, string renu_detail,
string renu_tsp) {
    count + +;
    renu[count].id = id;
    renu[count].detail = detail;
    renu[count].tsp = tsp;
}
sign string n_key(renu_id, renu_detail,
renu_tsp);
verify string p_key(renu_id, renu_detail,
renu_tsp);
```

```
function check smart cos tract renuchk(
string renu_id, string renu_detail,
string renu_tsp){
    return renu_id.exist;
    return renu_detail.exist;
    return renu_tsp.exist;
}
```

N uploads the resulting (ID_{BC}, BC_N) to the blockchain center by computing:

$$BC_N = h_9(r_N, s_N), \tag{81}$$

Step 5: N generates a random value k_{10}, and computes:

$$z_N = h_{10}(ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC}), \tag{82}$$

and execute the function "Sign" shown in Algorithm 4 to generate the signature:

$$Sign(z_N, k_{10}, d_N), \tag{83}$$

$$C_{10} = E_{PK_P}(ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC}). \tag{84}$$

Sends $ID_N, ID_P, C_{10}, (r_N, s_N)$ to the digital trading card agent, where M_{N_5} is the N sent a permission message to P.

Step 6: Digital trading card proxy first calculates:

$$(ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC}) = D_{SK_P}(C_{10}), \tag{85}$$

and calculates:

$$TS_{NOW} - TS_N \leq \Delta T. \tag{86}$$

To verify the validity of the timestamp, determine the correctness of the ECDSA signature, and then calculate:

$$z_N{}' = h_{10}(ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC}), \tag{87}$$

and use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_N{}', r_N, s_N). \tag{88}$$

If the authentication is passed, P will obtain N's permission to browse to the user and trigger the smart contracts renpins and renpchk; Algorithm 13 is as follows:

P uploads the obtained (ID_{BC}, BC_N) to the blockchain center by computing:

$$BC_N = h_{10}(r_N, s_N), \tag{89}$$

Algorithm 13. The scheme of the smart contracts renpins and renpchk

```
function insert smart constract renpins(
string renp_id, string renp_detail,
string renp_tsp) {
    count + +;
    renp[count].id = id;
    renp[count].detail = detail;
    renp[count].tsp = tsp;
}
sign string n_key(renp_id, renp_detail,
renp_tsp);
verify string p_key(renp_id, renp_detail,
renp_tsp);
```

```
function check smart cos tract renpchk(
string renp_id, string renp_detail,
string renp_tsp){
    return renp _id.exist;
    return renp _detail.exist;
    return renp _tsp.exist;
}
```

Digital trading card proxy starts issuing digital trading cards and agent keys to users, calculating:

$$c_1{}' = c_1{}^{(b/a)}, \tag{90}$$

$$z_P = h_{11}(ID_P, ID_U, AC, M_{P_1}, c_1{}', c_2, TS_P, ID_{BC}), \tag{91}$$

and executes the function "Sign" as shown in Algorithm 2 to generate the signature:

$$(r_P, s_P) = Sign(z_P, k_{11}, d_P), \tag{92}$$

$$C_{11} = E_{PK_U}(ID_P, ID_U, AC, M_{P_1}, c_1{}', c_2, TS_P, ID_{BC}). \tag{93}$$

Sends $ID_P, ID_U, C_{11}, (r_P, s_P)$ to the user, where M_{P_1} is information about the user's access to the digital trading card.

Step 7: The user first calculates:

$$(ID_P, ID_U, AC, M_{P_1}, c_1{}', c_2, TS_P, ID_{BC}) = D_{SK_U}(C_{11}), \tag{94}$$

and calculates:

$$TS_{NOW} - TS_P \leq \Delta T, \tag{95}$$

to verify the validity of the timestamp and check the correctness of the ECDSA signature.

$$z_P{}' = h_{10}(ID_P, ID_U, AC, M_P, c_1{}', c_2, TS_P, ID_{BC}). \tag{96}$$

Then, use the "Verify" function in Algorithm 4 to verify the signature:

$$Verify(z_P{}', r_P, s_P). \tag{97}$$

If the verification is passed, the user will obtain the information sent by the digital trading card agent and trigger the smart contracts repuins and repuchk, which are as follows (Algorithm 14).

The user uploads the obtained (ID_{BC}, BC_P) to the blockchain center by calculating:

$$BC_P = h_{11}(r_P, s_P). \tag{98}$$

Finally, the user computes:

$$ID_{DC} = c_1{}'^{(b)}/c_2. \tag{99}$$

Algorithm 14. The scheme of the smart contracts repuins and repuchk

```
function insert smart constract repuins(
string repu_id, string repu_detail,
string repu_tsp, string repu_ac) {
    repu[count].id = id;
    repu[count].detail = detail
    repu[count].tsp = tsp;
    repu[count].ac = ac;
sign string p_key(repu_id, repu_detail,
repu_tsp, string repu_ac);
verify string a_key(repu_id, repu_detail,
repu_tsp, string repu_ac);
```

```
function check smart cos tract repuchk(
string repu_id, string repu_detail,
string repu_tsp, string repu_ac ){
    return repu _id.exist;
    return repu _detail.exist;
    return repu _tsp.exist;
    return repu _ac.exist;
}
```

And the digital trading card is obtained. The digital trading card agent then uses ID_{DC} and obtains the $Cert_U$ and $VTime_U$ through ID_{DC} to be able to access the contents of the digital trading card at the specified time using the player.

4. Security Analysis

4.1. Dispersive and Membership Access Control

In the proposed scheme of this system, the blockchain network of the Hyperledger Fabric platform is used to build the consortium chain. The real entities correspond to the various organizations and peer nodes in the consortium chain. These organizations and peer nodes first need to register with the blockchain center and be recognized by the certificate authority to join the blockchain network and the corresponding channels. The peer nodes within the same organization belong to one organization and can trust each other, and the trust between different organizations is achieved by the certificate authority's authentication. Information can be shared openly and transparently between entities by joining the same channel interaction, and information can be isolated from other entities through the channel. Using this model, we can build a decentralized system with transparent information security and mutual trust between different organizations.

4.2. Data Integrity and Unforgeable Data

In the proposed scheme, an ECDSA-based blockchain platform is used, and the parties need to store the messages on the chain using private key signatures. In the data sending stage, the sender takes the hash value of the message and signs it, using the ECDSA algorithm, and the attacker cannot tamper with the data, thus ensuring the data integrity. For example, in the stage of the manufacturer obtaining the manufacturing authorization, the digital trading card manufacturer M sends a message $M_1 = ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}$ to the digital trading card center N. The manufacturer needs to generate the corresponding $z_M = h(M_1)$, and then performs the ECDSA algorithm on the hash value to achieve the signature $(r_M, s_M) = Sign(z_M, k_1, d_M)$. The signatures of each stage are shown in Table 2.

4.3. Transparency

All the information on digital trading cards is publicly uploaded, and anyone who joins the blockchain can verify the correctness of the uploaded information they want to query, and the information is open and transparent. In addition, this paper uses a proxy re-encryption algorithm for browsing digital trading cards, and the identity of the authorized object of browsing digital trading cards is verified by the digital trading card center. The time limit for browsing the token of the digital trading card is decided by the digital trading card center so the user cannot take possession of the digital trading card. We use blockchain to issue certificates to the entities that need to join the channel through a CA, making it possible for all nodes participating in the system to reach trust cost-effectively. The consensus mechanism of the blockchain makes the information stored in each node

consistent, making it impossible for entities to cheat each other. For example, when a user (U) needs to browse for access to a digital transaction card, it needs to apply to the digital transaction card center (N). The digital transaction card center (N) issues a key to the user (U) that contains U's certificate $Cert_U$ and a valid time $VTime_U$ for the user.

Table 2. Verification of data integrity and unforgeability of the proposed scheme.

Phase	Party		Message	Hash Value	Verification
	Sender	Receiver			
Manufacturing	M	N	$M_1 = ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}$	$z_M = h(M_1)$	$(r_M, s_M) = Sign(z_M, k_1, d_M)$
Authorization	N	M	$M_2 = ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC}$	$z_N = h(M_2)$	$(r_N, s_N) = Sign(z_N, k_2, d_N)$
Review	M	N	$M_3 = ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC}$	$z_M = h(M_3)$	$(r_M, s_M) = Sign(z_M, k_3, d_M)$
Issued	N	P	$M_4 = ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC}$	$z_N = h(M_4)$	$(r_N, s_N) = Sign(z_N, k_4, d_N)$
Verify User Id	U	N	$M_5 = ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC}$	$z_U = h(M_5)$	$(r_U, s_U) = Sign(z_U, k_5, d_U)$
and Invoice	N	U	$M_6 = ID_N, ID_U, M_{N_3}, TID, invoice, TS_N, ID_{BC}$	$z_N = h(M_6)$	$(r_N, s_N) = Sign(z_N, k_6, d_N)$
Payment	B	U	$M_7 = ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC}$	$z_B = h(M_7)$	$(r_B, s_B) = Sign(z_B, k_7, d_B)$
Browse	U	N	$M_8 = ID_U, ID_N, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC}$	$z_U = h(M_8)$	$(r_U, s_U) = Sign(z_U, k_8, d_U)$
Request	N	U	$M_9 = ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC}$	$z_N = h(M_9)$	$(r_N, s_N) = Sign(z_N, k_9, d_N)$
and	N	P	$M_{10} = ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC}$	$z_N = h(M_{10})$	$(r_N, s_N) = Sign(z_N, k_{10}, d_N)$
Access	P	U	$M_{11} = ID_P, ID_U, AC, M_{P_1}, c_1', c_2, TS_P, ID_{BC}$	$z_P = h(M_{11})$	$(r_P, s_P) = Sign(z_P, k_{11}, d_P)$

4.4. Traceability

The design uses the Hyperledger Fabric blockchain platform, which is a consortium chain platform where only entities recognized by the certificate authority can join the blockchain and use the corresponding public key to verify the data. This method is used to achieve traceability of data on the blockchain, prevent digital information forgery, and ensure the fairness of the digital trading card market. For example, if the digital trading card center (N) wants to trace the audit information of the digital trading card manufacturer (M), the digital trading card center (N) needs to use the public key Q_M of the digital trading card manufacturer (M) and use $Verify(z_M', r_M, s_M)$ to verify that the message was indeed uploaded by M, and M needs to be responsible for the uploaded information. Therefore, we achieve traceability of the message. The traceability verification at each stage is shown in Table 3 below.

Table 3. Traceability verification at each stage.

Phase	Party		Verification
	Sender	Receiver	
Manufacturing	M	N	$Verify(z_M', r_M, s_M)$
Authorization	N	M	$Verify(z_N', r_N, s_N)$
Review	M	N	$Verify(z_M', r_M, s_M)$
Issued	N	P	$Verify(z_N', r_N, s_N)$
Verify User Id	U	N	$Verify(z_U', r_U, s_U)$
and Invoice	N	U	$Verify(z_N', r_N, s_N)$
Payment	B	U	$Verify(z_B', r_B, s_B)$
Browse	U	N	$Verify(z_U', r_U, s_U)$
Request	N	U	$Verify(z_N', r_N, s_N)$
and	N	P	$Verify(z_N', r_N, s_N)$
Access	P	U	$Verify(z_P', r_P, s_P)$

4.5. Timeless

In the proposed scheme, the digital trading card center (N) is responsible for the identity and payment certificate of the user (U), and then the digital trading card center licenses the digital trading card agent to issue a time-limited broadcast license key to the

user, who cannot achieve permanent digital trading card access through a single browsing application and spreads the browsing authorization to others at will. In this way, we avoid the leakage of digital transaction cards.

4.6. Replay Attacks

In proceeding with a message transmission, the message passes through an insecure transmission channel. An attacker can compromise the authentication of the system by intercepting legitimate message packets sent from sender A to receiver B. The attacker uses the information intercepted from the last communication content to impersonate sender A [35]. We add timestamps to each message to prevent replay attacks. For example, in the data transmission phase, the user (U) sends a message M_U to the digital trading card center (N). It is necessary to add a timestamp TS_U to it, and N determines whether the message was intercepted by a third party by verifying the timestamp $TS_{NOW} - TS_U \leq \Delta T$. Even if the third party tampers with a timestamp TS_U, we can determine whether the timestamp has been tampered with by using algorithm 4.

4.7. Man-in-the-Middle Attack

We also use asymmetric encryption for communication messages to resist the interception and tampering of message contents by attackers. For example, the message sender user (U) can look up the public key of the receiver's digital trading card center (N) from the blockchain network and use N's public key to asymmetrically encrypt. N receives the message and decrypts C_5 with his private key as follows $(ID_U, ID_N, M_U, Cert_U, TS_U, ID_{BC}) = D_{SK_N}(C_5)$. The attacker of the message does not know the private key of the receiver N and therefore cannot decrypt the message. Table 4 below shows the asymmetric encryption and decryption process of the communication message in each stage.

Table 4. The asymmetric encryption and decryption process of the communication message in each stage.

Phase	Party		Encryption	Decryption
	Sender	Receiver		
Manufacturing	M	N	$C_1 = E_{PK_N}(ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC})$	$(ID_M, ID_N, M_{M_1}, Cert_M, TS_M, ID_{BC}) = D_{SK_N}(C_1)$
Authorization	N	M	$C_2 = E_{PK_M}(ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC})$	$(ID_N, ID_M, M_{N_1}, AC, TS_N, ID_{BC}) = D_{SK_M}(C_2)$
Review	M	N	$C_3 = E_{PK_N}(ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC})$	$(ID_M, ID_N, M_{M_2}, Cert_M, AC, key_m, TS_M, ID_{BC}) = D_{SK_N}(C_3)$
Issued	N	P	$C_4 = E_{PK_P}(ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC})$	$ID_N, ID_P, M_{N_2}, Cert_N, AC, c_1, c_2, key_m, TS_N, ID_{BC}) = D_{SK_P}(C_4)$
Verify User Id	U	N	$C_5 = E_{PK_N}(ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC})$	$(ID_U, ID_N, M_{U_1}, Cert_U, TS_U, ID_{BC}) = D_{SK_N}(C_5)$
and Invoice	N	U	$C_6 = E_{PK_U}(ID_N, ID_U, M_{N_3}, TID, invoice, TS_N, ID_{BC})$	$(ID_N, ID_U, M_{N_3}, TID, invoice, TS_N, ID_{BC}) = D_{SK_U}(C_6)$
Payment	B	U	$C_7 = E_{PK_U}(ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC})$	$(ID_B, ID_U, M_{B_1}, Cert_{pay}, TS_B, ID_{BC}) = D_{SK_U}(C_7)$
Browse	U	N	$C_8 = E_{PK_N}(ID_U, ID_N, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC})$	$(ID_U, ID_N, M_{U_2}, Cert_U, TID, Cert_{pay}, TS_U, ID_{BC}) = D_{SK_N}(C_8)$
Request	N	U	$C_9 = E_{PK_N}(ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC})$	$(ID_N, ID_P, b, M_{N_4}, TS_N, ID_{BC}) = D_{SK_A}(C_9)$
and	N	P	$C_{10} = E_{PK_P}(ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC})$	$(ID_N, ID_P, M_{N_5}, b/a, TS_N, ID_{BC}) = D_{SK_P}(C_{10})$
Access	P	U	$C_{11} = E_{PK_U}(ID_P, ID_U, AC, M_{P_1}, c_1', c_2, TS_P, ID_{BC})$	$(ID_P, ID_U, AC, M_{P_1}, c_1', c_2, TS_P, ID_{BC}) = D_{SK_U}(C_{11})$

4.8. Sybil Attack

A Sybil attack is a type of attack against P2P networks, where the attacker attacks the redundancy of the blockchain network by simulating the creation of multiple identities to disrupt the consensus mechanism and thus take control of the network. In the Hyperledger Fabric used in our system, a certificate authority CA is used to authorize the nodes joining the network, and all the nodes are managed using certificates to avoid witch attacks.

5. Discussion

5.1. Computation Cost

In this section, we analyze the computational cost for each role in each phase of the study. We use asymmetric encryption and decryption, ECDSA signature and verification functions, hashing operations, symmetric encryption operations, multiplication, and division operations as the basis for calculating the costs. The cost of each phase is shown in Table 5.

Table 5. Computation costs of the proposed scheme.

Phase	1st Role	2nd Role	3rd Role
Manufacturing Authorization	Digital Trading Cards Manufacturer	NBA Digital Trading Cards Center	N/A
	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D}$	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D}$	
Review	Digital Trading Cards Manufacturer	NBA Digital Trading Cards Center	N/A
	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D} + T_{Sym}$	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D} + T_{Sym}$	
Issued	NBA Digital Trading Cards Center	Digital Trading CardsProxy	N/A
	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D} + T_{\exp} + 2T_{div} + T_{mul}$	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D}$	
Verify User Id and Invoice	User	NBA Digital Trading Cards Center	N/A
	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D}$	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D}$	
Payment	Bank	User	N/A
	$T_{sig} + T_{hash} + T_{E/D}$	$T_{ver} + 2T_{hash} + T_{E/D}$	
Browse Request and Access	User	NBA Digital Trading Cards Center	Digital Trading CardsProxy
	$T_{sig} + 2T_{ver} + 5T_{hash} + 3T_{E/D} + T_{\exp} + T_{div}$	$2T_{sig} + T_{ver} + 4T_{hash} + 3T_{E/D}$	$T_{sig} + T_{ver} + 3T_{hash} + 2T_{E/D} + T_{\exp} + T_{div}$

T_{sig}: signature operation; T_{ver}: verify operation; $T_{E/D}$: encryption/decryption operation; T_{hash}: hash function operation; $T_{E/D}$: encryption/decryption operation; T_{Sym}: symmetric encryption/decryption operation; $T_{\exp}$: an exponential operation; T_{div}: a division operation; T_{mul}: a multiplication operation.

5.2. Communication Cost

In Table 6, the communication cost of each phase in this scheme is analyzed. The maximum transmission speed is 14 Mbps in a 3.5G communication environment, 100 Mbps in a 4G communication environment, and 20 Gbps in a 5G communication environment. In this analysis, we assume that the ECDSA signature is 160 bits, asymmetrically encrypted messages are 1024 bits, symmetrically encrypted messages are 512 bits, and other message lengths (such as ID) are 80 bits. For example, in the manufacturing authorization phase, the amount of communication data between the NBA digital trading card center (N) and the digital trading card manufacturer (M) is four ID messages, two asymmetric encryption letters, and two signature messages, i.e., 4 * 80 bits + 2 * 1024 bits + 2 * 160 = 2688 bits total communication overhead for this phase. It takes 0.192 ms in a 3.5G communication environment, 0.027 ms in a 4G communication environment, and 0.134 µs in a 5G communication environment. We think this scheme has excellent performance.

Table 6. Computation costs of the proposed scheme.

Phase	Message Length	3.5G (14 Mbps)	4G (100 Mbps)	5G (20 Gbps)
Manufacturing Authorization	2688 bits	0.192 ms	0.027 ms	0.134 µs
Review	1856 bits	0.133 ms	0.019 ms	0.093 µs
Issued	1362 bits	0.097 ms	0.014 ms	0.068 µs
Verify User Id and Invoice	2688 bits	0.192 ms	0.027 ms	0.134 µs
Payment	1344 bits	0.096 ms	0.014 ms	0.065 µs
Browse and Access	5076 bits	0.384 ms	0.054 ms	0.268 µs

5.3. Performance Analysis-Deploying System Based on Blockchain

In this section, we describe the deployment of the proposed method. Hyperledger Fabric uses Docker container technology to build the network. The Fabric network includes order nodes, CA nodes, and peer nodes. The scenario is described as follows: Peer nodes represent the various entities participating in the blockchain network. Each node has a blockchain data ledger and runs on a Docker container. We deploy the test network of fabric samples officially provided by Fabric. Hyperledger caliper version 0.42 and the blockchain platform uses Hyperledger Fabric version 2.3, on a server with an AMD R7 5800H@3.2 GHz CPU and 4 GB RAM.

The following Figures 12 and 13 have illustrated the throughput and latency of smart contract calling.

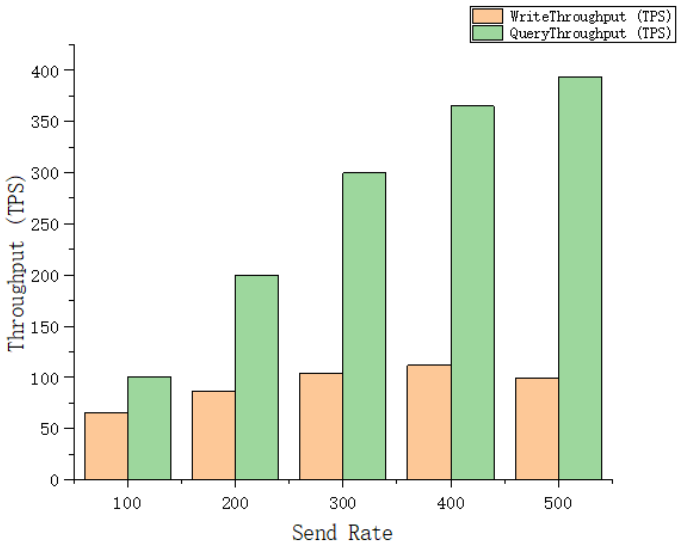

Figure 12. Throughput of transaction with varying Send Rate.

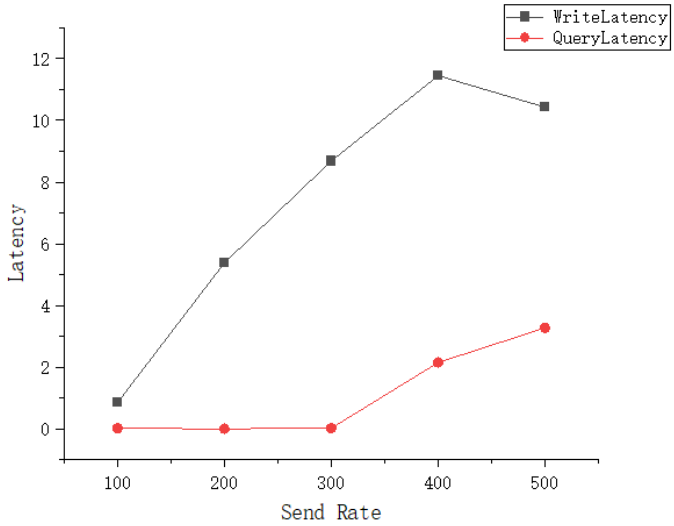

Figure 13. Latency of transaction with varying Send Rate.

In Figure 12, we select the contract with the largest amount of data on the chain as the object of our performance test. By adjusting the Send Rate, we recorded the write and query throughput under the Send Rate of 100 to 500. WriteThroughput and QueryThroughput increase as Send Rate increases. When the Send Rate reaches 400, the written workload reaches the bottleneck and cannot be increased anymore.

In Figure 13, QueryLatency increases significantly when Send Rate reaches 300. Meanwhile, WriteLatency maintains an upward trend from Send Rate 100 to 400 and then decreases at Send Rate 500.

5.4. Function Comparison

Some of the systems mentioned in the related work on digital content are compared in Table 7 below.

Table 7. Comparison between Ethereum and Hyperledger Fabric.

Authors	Year	Objectives	1	2	3	4	5	6
Ma et al. [21]	2018	Blockchain-based scheme for digital rights management	Y	Y	Y	Y	Y	N
Ma et al. [22]	2019	Permissioned blockchain-based decentralized trust management and secure usage control scheme of IoT big data	Y	N	N	Y	N	N
Guo et al. [23]	2020	Combination of the public and private blockchains-enabled digital rights management system	Y	N	N	Y	N	N
Khan et al. [24]	2020	Content protection and transaction method using blockchain Ethereum technology	Y	N	N	Y	N	N
Li et al. [25]	2021	Blockchain-watermarking scheme to protect the privacy	Y	Y	N	Y	N	N
Heo et al. [26]	2021	Using SBBC's blockchain digital content trading system	Y	N	N	Y	N	N
Ours	2022	Blockchain-based digital trading card management platform	Y	Y	Y	Y	Y	Y

Notes: 1: Blockchain-focused, 2: Comparative analysis with other approaches using tables, 3: Security Analysis, 4: Unforgeable, 5: Traceable, 6: Timeless, Y: Yes and N: No.

In general, when we compare the six indicators of the blockchain-focused, comparative analysis with other approaches using tables, security analysis, unforgeable, traceable, and timeless in other digital property articles, only the proposed scheme can meet the above indicators.

6. Conclusions

The issue of NBA digital trading card anti-counterfeiting and its information is closely related to collectors. To solve the information asymmetry problem between buyers and sellers of digital trading cards and guarantee the security of digital trading card ownership, we propose a blockchain-based NBA digital trading card management system with anti-counterfeiting traceability. The production and issuance of each NBA digital trading card cannot pass the authorization and audit of the NBA digital trading card management center and be stored on the chain. The NBA digital trading card center as a seller cannot tamper with the selling information either. The blockchain can also provide practical and powerful evidence to protect collectors' rights and interests if a dispute arises over the ownership of digital trading cards. We propose a digital trading card supply chain framework based on blockchain and smart contracts. To build a more secure system, we apply the ECDSA in the communication protocol, and we analyze the security of the system, such as the data integrity and tamper-evident data, distributed and member access, information transparency, and traceability. In addition to this, we believe that the proposed protocol is resistant to man-in-the-middle attacks, replay attacks, and witch attacks. Then, the proposed scheme is discussed in terms of the computational cost and the communication cost and compared with other schemes. Our system can protect the digital content more securely. In summary, we obtain the following contributions:

(1) Hyperledger Fabric technology is used for the management of anti-counterfeit traceability of digital trading cards.
(2) The ECDSA signature algorithm can be used to ensure data integrity.
(3) Proxy re-encryption is used for secure and trusted digital content authorization.
(4) Smart contracts are designed for the ins and outs of digital transaction cards.
(5) An analysis of the computational and communication costs.
(6) Collectors can verify the ownership of NBA digital trading cards via blockchain.

Author Contributions: Conceptualization, C.-L.C. and C.-C.F.; methodology, C.-L.C., C.-C.F. and M.Z.; validation, W.-J.T., W.Z. and H.S.; investigation, C.-L.C. and C.-C.F.; data analysis, W.-J.T., W.Z. and H.S.; writing—original draft preparation, C.-L.C. and C.-C.F.; writing—review and editing, W.-J.T., Y.-Y.D. and H.S.; supervision, C.-L.C. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Science and Technology Project of Jilin Provincial Department of Education (JJKH20210457KJ), the Undergraduate Training Programs for Innovation and Entrepreneurship Project of Jilin Province (J202210203JSJ02), and 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: Not applicable.

Informed Consent Statement: Not applicable.

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.

Notations:

q	A k-bit prime number
$GF(q)$	Finite group q
E	The elliptic curve defined on finite group q
G	A generating point based on the elliptic curve E
ID_X	A name representing an identity X
k_i	The i-th generated a random number on the elliptic curve
AC	N generated digital cards to generate authorization codes
$(r_X, s_X)/(x_X, y_X)$	The i-th ECDSA/elliptic curve signature value of user X
d_X/Q_X	Private/public key of X
a/b	Private/public key in proxy re-encryption
M_{X_i}	The i-th message sent by role X
ID_{BC}	An index value of blockchain message
BC_X	Blockchain messages for X
$TS_X/TS_{NOW}T_i$	X's timestamp/current timestamp
PK_X/SK_X	X's public key/private key that was issued by BBC
$E_{PK_X}(M)/D_{SK_X}(M)$	Encrypt/decrypt the message M using X's public key/private key
$E_{key_m}(M)/D_{key_m}(M)$	Encrypt/decrypt the digital content using symmetric keys
ID_{DC}	An identity of digital content
key_m	Asymmetric key containing KeyID and Seed
$Cert_X$	A digital certificate X conforms to the X.509 standard
$Cert_{Pay}$	Certification issued by the bank for the user's purchase
Z_X	The hash value of X
$h_i()$	The i-th hash function
C_{sym}	The ciphertext obtained by symmetric encryption
$x_X \stackrel{?}{=} r_X.$	Verify whether x_X is equal to r_X

References

1. Trading Card—Wikipedia. Available online: https://en.wikipedia.org/wiki/Trading_card (accessed on 22 July 2022).
2. Sports Trading Card Market Size and Forecast. Available online: https://www.verifiedmarketresearch.com/product/sports-trading-card-market/ (accessed on 22 July 2022).

3. Ebay's 2021 "State of Trading Cards" Report Spotlights Collecting Trends and Industry Predictions. Available online: https://www.ebayinc.com/stories/news/ebays-2021-state-of-trading-cards-report-spotlights-collecting-trends-and-industry-predictions/ (accessed on 22 July 2022).
4. Operation Bullpen. Available online: https://archives.fbi.gov/archives/news/stories/2005/july/operation-bullpen-overview (accessed on 22 July 2022).
5. Rovell: A Look Inside the Scandal Rocking the Sports Memorabilia World. Available online: https://www.actionnetwork.com/general/darren-rovell-card-memorabilia-fraud-national-sports-collector-convention (accessed on 22 July 2022).
6. PSA-Wikipedia. Available online: https://en.wikipedia.org/wiki/PSA (accessed on 22 July 2022).
7. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260. Available online: https://klausnordby.com/bitcoin/Bitcoin_Whitepaper_Document_HD.pdf (accessed on 22 July 2022).
8. Ante, L. The non-fungible token (NFT) market and its relationship with Bitcoin and Ethereum. *FinTech* **2022**, *1*, 216–224. [CrossRef]
9. Chen, C.-L.; Lin, C.-Y.; Chiang, M.-L.; Deng, Y.-Y.; Chen, P.; Chiu, Y.-J. A traceable online will system based on blockchain and smart contract technology. *Symmetry* **2021**, *13*, 466. [CrossRef]
10. Chen, C.-L.; Deng, Y.-Y.; Tsau, W.-J.; Li, C.-T.; Lee, C.-C.; Wu, C.-M. A traceable online insurance claims system based on blockchain and smart contract technology. *Sustainability* **2021**, *13*, 9386. [CrossRef]
11. Chen, C.; Lim, Z.; Liao, H.; Deng, Y.; Chen, P. A traceable and verifiable tobacco products logistics system with GPS and RFID technologies. *Appl. Sci.* **2021**, *11*, 4939. [CrossRef]
12. Deepa, N.; Pham, Q.; Nguyen, D.; Bhattacharya, S.; Prabadevi, B.; Gadekallu, T.; Maddikunta, P.; Fang, F.; Pathirana, P. A survey on blockchain for big data: Approaches, opportunities, and future directions. *Future Gener. Comput. Syst.* **2022**, *131*, 209–226. [CrossRef]
13. Stamatellis, C.; Papadopoulos, P.; Pitropakis, N.; Katsikas, S.; Buchanan, W.J. A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric. *Sensors* **2020**, *20*, 6587. [CrossRef]
14. Chen, C.; Wang, T.; Tsaur, W.; Weng, W.; Deng, Y.; Cui, J. Based on Consortium Blockchain to Design a Credit Verifiable Cross University Course Learning System. *Secur. Commun. Netw.* **2021**, *2021*, 8241801. [CrossRef]
15. Chen, C.-L.; Deng, Y.-Y.; Li, C.-T.; Zhu, S.; Chiu, Y.-J.; Chen, P.-Z. An IoT-Based Traceable Drug Anti-Counterfeiting Management System. *IEEE Access* **2020**, *8*, 224532–224548. [CrossRef]
16. Zhao, S.; O'Mahony, D. Bmcprotector: A blockchain and smart contract based application for music copyright protection. In Proceedings of the 2018 International Conference on Blockchain Technology and Application, Xi'an, China, 10–12 December 2018; pp. 1–5.
17. Jing, N.; Liu, Q.; Sugumaran, V. A blockchain-based code copyright management system. *Inf. Process. Manag.* **2021**, *58*, 102518. [CrossRef]
18. Wang, Y.-C.; Chen, C.-L.; Deng, Y.-Y. Authorization Mechanism Based on Blockchain Technology for Protecting Museum-Digital Property Rights. *Appl. Sci.* **2021**, *11*, 1085. [CrossRef]
19. Wang, N.; Xu, H.; Xu, F.; Cheng, L. The algorithmic composition for music copyright protection under deep learning and blockchain. *Appl. Soft Comput.* **2021**, *112*, 107763. [CrossRef]
20. NBA Top Shot Mints A Unicorn: How an Ethereum Competitor Cashed in on The NFT Craze. Available online: https://www.forbes.com/sites/brettknight/2021/03/30/nba-top-shot-dapper-labs-nft-funding/?sh=2b0b234a67ae (accessed on 22 July 2022).
21. Ma, Z.; Jiang, M.; Gao, H.; Wang, Z. Blockchain for digital rights management. *Future Gener. Comput. Syst.* **2018**, *89*, 746–764. [CrossRef]
22. Ma, Z.; Wang, L.; Wang, X.; Wang, Z.; Zhen, W. Blockchain-enabled decentralized trust management and secure usage control of IoT big data. *IEEE Internet Things J.* **2019**, *7*, 4000–4015.
23. Guo, J.; Li, C.; Zhang, G.; Sun, Y.; Bie, R. Blockchain-enabled digital rights management for multimedia resources of online education. *Multimed. Tools Appl.* **2020**, *79*, 9735–9755. [CrossRef]
24. Khan, U.; An, Z.; Imran, A. A blockchain Ethereum technology-enabled digital content: Development of trading and sharing economy data. *IEEE Access* **2022**, *8*, 217045–217056. [CrossRef]
25. Li, M.; Zeng, L.; Zhao, L.; Yang, R.; An, D.; Fan, H. Blockchain-watermarking for compressive sensed images. *IEEE Access* **2021**, *9*, 56457–56467. [CrossRef]
26. Heo, G.; Yang, D.; Doh, I.; Chae, K. Efficient and secure blockchain system for digital content trading. *IEEE Access* **2021**, *9*, 77438–77450. [CrossRef]
27. Polge, J.; Robert, J.; Le Traon, Y. Permissioned blockchain frameworks in the industry: A comparison. *ICT Express* **2020**, *7*, 229–233. [CrossRef]
28. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
29. Gai, R.; Du, X.; Ma, S.; Chen, N.; Gao, S. A summary of the research on the foundation and application of blockchain technology. *J. Phys. Conf. Ser.* **2020**, *1693*, 012025. [CrossRef]
30. Nasir, Q.; Qasse, I.A.; Abu Talib, M.; Nassif, A. Performance analysis of hyper ledger fabric platforms. *Secur. Commun. Netw.* **2018**, *2018*, 3976093. [CrossRef]

31. Chen, C.-L.; Yang, J.; Tsaur, W.-J.; Weng, W.; Wu, C.-M.; Wei, X. Enterprise Data Sharing with Privacy-Preserved Based on Hyperledger Fabric Blockchain in IIOT's Application. *Sensors* **2022**, *22*, 1146. [CrossRef] [PubMed]
32. Szabo, N. The Idea of Smart Contracts. Nick Szabo's Papers and Concise Tutorials. 1997, 6, p. 199. Available online: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html (accessed on 22 July 2022).
33. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
34. Blaze, M.; Bleumer, G.; Strauss, M. Divertible protocols and atomic proxy cryptography. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin, Germany, 1998; pp. 127–144.
35. Malladi, S.; Alves-Foss, J.; Heckendorn, R.B. On Preventing Replay Attacks on Security Protocols. 2002. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.6492&rep=rep1&type=pdf (accessed on 22 July 2022).