applied sciences

MDPI

Article

Authorization Mechanism Based on Blockchain Technology for Protecting Museum-Digital Property Rights

Yun-Ciao Wang [1], Chin-Ling Chen [2,3,4,*] and Yong-Yuan Deng [2,*]

[1] National Museum of Marine Biology and Aquarium, Pingtung 94450, Taiwan; yunciao@nmmba.gov.tw
[2] Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
[3] School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China
[4] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361005, China
* Correspondence: clc@mail.cyut.edu.tw (C.-L.C.); allen.nubi@gmail.com (Y.-Y.D.)

Featured Application: Museums not only achieve the goal of promoting social education, but also solve their financial problems.

Abstract: In addition to the exhibition, collection, research, and educational functions of the museum, the development of a future museum includes the trend of leisure and sightseeing. Although the museum is a non-profit organization, if it can provide digital exhibits and collections under the premises of "intellectual property rights" and "cultural assets protection", and licensing and adding value in various fields, it can generate revenue from digital licensing and handle the expenses of museum operations. This will be a new trend in the sustainable development of museum operations. Especially since the outbreak of COVID-19 at the beginning of this year (2020), the American Alliance of Museums (AAM) recently stated that nearly a third of the museums in the United States may be permanently closed since museum operations are facing "extreme financial difficulties." This research is aimed at museums using the business model of "digital authorization". It proposes an authorization mechanism based on blockchain technology protecting the museums' digital rights in the business model and the application of cryptography. The signature and time stamp mechanism achieve non-repudiation and timeless mechanism, which combines blockchain and smart contracts to achieve verifiability, un-forgery, decentralization, and traceability, as well as the non-repudiation of the issue of cash flow with signatures and digital certificates, for the digital rights of museums in business. The business model proposes achievable sustainable development. Museums not only achieve the goal of promoting social education, but also solve their financial problems.

Keywords: museum; digital copyright management; blockchain; smart contract; authorization model

Citation: 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. https://doi.org/10.3390/app11031085

Received: 23 December 2020
Accepted: 21 January 2021
Published: 25 January 2021

1. Introduction

In addition to their exhibition, collection, research, and education functions, museums' main purpose is to display and protect cultural resources. Continuous attention has been paid to them. However, there is a difficulty: If these collections are displayed in public places for a long time, they may deteriorate. On the other hand, if they are kept in a warehouse, visitors cannot share this valuable information. In 2007, Ross Parry suggested that the concept of digital collections should be added to the main concepts of museums [1]. The main purpose is to digitize these collections. Moreover, transforming collections into digital content in a unified format and developing them into good digital rights management will not only help promote social education, but also facilitate the operation of museums.

Museum digitalization means that the museum converts the texts, images, and videotapes through digital scanners and digital cameras based on the collections of the museum to produce digital data that can be processed by a computer; "Digital Collection" refers

to the data and files on various utensils, paintings, and calligraphy, specimens, and documents that have been processed through digital processes. The "digital collections" authorization originated at the beginning of photography in the 19th century. The British Museum accepted donations of photographic images, as well as professional photographers' photo collections, and sold the collections taken in the museum and the records of museum activity photos; this was the beginning of the museum's image recording and image authorization [2].

In the 20th century, international museums and governments implemented the digitization plan of various museum collections based on the mission of collection preservation and promotion of cultural policies. Currently, in the 21st century, digital technology is booming, and museums have entered the era of digitization. There are a huge number of digital images. Production allows museums to hold numerous copyrights, signaling an important turning point for image authorization. The international museum community has invested a lot of money and human resources starting more than ten years ago to digitize its collections on a large scale. For example, J. Paul Getty Trust, associated with the Getty Museum, paid US$4.2 million from 1997 to 2002 in funding to establish the "Electronic Cataloguing Initiative", which sponsored 21 Los Angeles area museums whose main collections are visual arts.

In 2009, the foundation launched OSCI in cooperation with the J. Paul Getty Museum and eight other institutions. Arthur M. Sackler and Freer Art Gallery; Los Angeles County Museum of Art; National Gallery of Art in Washington, DC; San Francisco Museum of Modern Art; Seattle Art Museum; and the Tate and Walker Art Center. The goal of the alliance is to create models for online catalogs, which will greatly increase access to museum collections to provide interdisciplinarity and the latest research, and innovate how to conduct, introduce, and use this research [3]. In 2002, the Culture Online Project of the British Department of Culture, Media and Sports was founded [4]; the British Museum established the "Merlin Project" in 2006 along with other projects, which are all efforts related to the museum's digital collection.

The core of the museum is its collection and heritage, the physical evidence of human survival and its environment. This includes two levels of connotation: One is the cultural relic entity in the museum's collection; the other is the information resources that recreate the cultural relic entity, reveal its original information and cultural connotation, including text introductions, images, video three-dimensional models, etc. Museum experts and scholars research and publish works on a certain collection or collection preservation technology, as well as works of collection pictures taken by museum photographers, etc., all belonging to the collection resources.

The "Creative Economy Report 2010" of UNESCO [4] points out that "cultural heritage" is the source of all art forms, and the soul of culture and creative industries, which brings together history, anthropology, ethnology, aesthetics, and social perspectives, while influences people's creativity. The intellectual property authorization of the museum means that the museum authorizes the copyright of its collection resources. It includes cultural relics, specimens, and artworks to other institutions for the development of cultural derivatives, transforms cultural resources into cultural goods, and establishes effective communication with consumers. It forms a unique brand of museums and reflects the intention of museums to develop products [5]. The authorized person pays the corresponding fee to the authorizer, and the authorizer gives the authorized person corresponding guidance and assistance. In particular, museums in various countries with rich collections can serve as models for brand authorization.

Brand authorization began in the United States in the early 20th century. When Disney's classic cartoon image of Mickey Mouse became famous, a furniture merchant paid Walt Disney US$300 in exchange for the right to print the image of Mickey Mouse on its products. Disney is recognized as the originator of international brand authorization. Currently, brand authorization has become a global industry with a relatively mature operation model and a complete industrial chain. According to the "2019 Global Licensing Industry

Market Survey Report" released by the International Licensing Industry Merchandiser's Association (LIMA) [6], the global retail sales of licensed goods reached 280.3 billion U.S. dollars in 2018, a year-on-year increase of 3.2%. Among the competitors, China's authorized industry market sales reached 9.5 billion U.S. dollars, maintaining a rapid growth trend with an increase of 67%.

As an image producer, the core mission of museums is to produce images in the spirit of equality, sharing, and reciprocity. This view also echoes the concept of equality of museums. In the comprehensive digital collection, most of the collections that cannot be displayed or watched in permanent exhibitions or special exhibitions can have the opportunity to be presented to the world. For example, the sea area around the National Museum of Marine Biology and Aquarium, located in the Kenting National Park in southern Taiwan, is a typical marine environment intersection, covering the estuary area, sandy mud bottom, reef shores, and other habitats. Chang et al. [7] studied and integrated the fish species in the sub-tidal zone around the National Museum of Marine Biology and Aquarium, which provides a constant monitoring and conservation research platform for the aquatic environment and biodiversity. The museum has also carried out the image management collection of collection resources [8], but how to use these valuable research resources of the museum through the appropriate preservation, management, authorization, and promote social education is an extremely important challenge.

In recent years, under the concept of "activating and reproducing collections", museums spread a huge amount of knowledge and culture to visitors with their rich collections, such as artworks, crafts, biological specimens, texts, drawings, paintings, photos, maps, movies, and sound recordings. Museums all over the world take marketization, digitization, diversification, and popularization as their development direction. Their development and utilization of digital image resources in the collections, via different authorization models, are widely praised by society.

The cultural industry chain is divided into four links: Research and development, production, circulation, and consumption. With the development over time, the term "authorization" has been widely used in the cultural industry, and its connotation and extension have also been continuously expanded, and gradually valued by museums. At present, there are two views on the definition of digital image authorization of museum collections: One view is that authorization refers to the process by which the museum grants the digital image of cultural relics owned or managed by the museum as the subject matter to the authorized person in the form of a contract; another view is that authorization is mainly the process of transaction and management of related intellectual property rights. The ultimate goal of museums' digital authorization of collections is to increase economic benefits based on spreading culture and exerting its educational function.

However, due to various reasons, most people may not be able to visit their favorite museums one by one due to time and space constraints. For example, since the outbreak of the COVID-19 at the beginning of this year (2020), the American Alliance of Museums (AAM) recently stated that nearly one-third of museums in the United States may be permanently closed, and pointed out that museum operations are facing "extreme financial difficulties" [9]. Therefore, determining how to protect museum collections and effectively use these collection resources to maintain the operation of the museum is a critical topic for consideration by museum operators.

Due to the fading of museum collections, while promoting social education, we must strive to preserve them. Digitizing collections is a feasible way. On the other hand, in order to maintain the sustainable operation of museums, it is important to manage the property rights of museum collections after digitization. Copyright provides a bridge between art and commerce because we need to protect the collections. In the past, using watermarking technology to achieve digital property management has been a mature technology [10–12]. Digital rights management is always inseparable from cryptographic technology [13–16]. Up to now, watermarks are combined with smart contract technology to realize digital property management [17]. In recent years, more scholars have used the

characteristics of decentralization, non-tampering, traceability, and blockchain openness to solve the application problems of digital rights management, a process that has expanded rapidly [17–22].

However, none of the above-mentioned digital property rights management mechanisms integrate the operation of the cash flow system, and naturally cannot reflect its feasibility. Therefore, this article integrates cash flow management into our digital rights management regarding comprehensive digital collections and promotes transparency of collections, the heart of museums. Apart from the practice of equality, the production of images provides an extension of museum collections and serves as a carrier of culture. The circulation of copied images creates richer and more diverse ways of use [23].

In 2017, Ma proposed a common, flexible, and extendable solution for variant DRM scenes, and can support rapid and customized development [24]. Du Toit proposed a decentralized architectural model, which makes use of digital rights management to enforce access control over personal information [25]. Mrabet et al. [26] concluded the open research issues and future directions towards securing IoT. Including the use of blockchain to address security challenges in IoT, and the implications of IoT deployment in 5G and beyond. Therefore, the first focus of digital rights management is how to achieve proper authorization. Generally, the authorization mode of digital collections in museums is divided into the following three methods:

1. Direct authorization model of museum digitized collections

The direct authorization model is a model in which the museum, as the authorized party, signs a contract with the authorized party to authorize it to use the digital resources of the collections. The museum collects cultural relics, produces digital content, encrypts and encapsulates, authorizes the identity verification and makes remittance notices, authorizes remittances royalties' feedback, and finally operates the key authorization process. The authorization model process is shown in Figure 1. The National Museum of the Netherlands and the British Museum, as well as the National Palace Museum in Taipei in Taiwan, are typical examples of the direct authorization model.

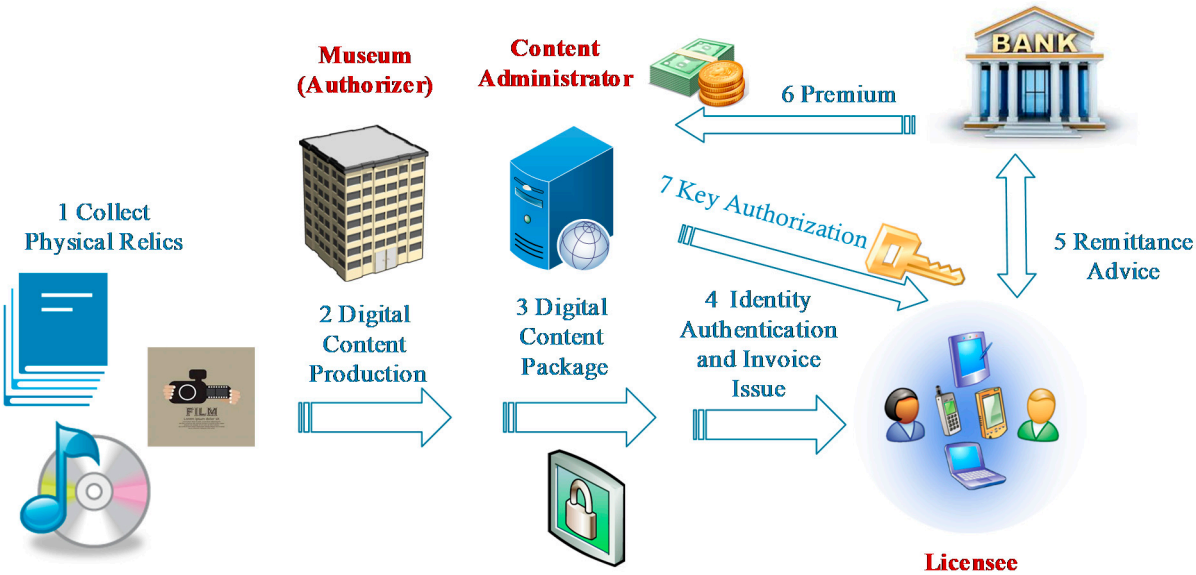

Figure 1. The digital direct authorization model of museum collections.

Under this authorization model, the authorized party often directly participates in the use of the digital image resources of cultural relics by third-party manufacturers. The advantage is that it is not only conducive to the museum as the authorized party to promptly understand the development of digital image resources, but is also given an in-depth understanding of the connotations of the collection by the relevant departments

of the museum, which is often helpful to the successful development of digital resources. However, the shortcomings of this authorization model are also obvious. Because the authorized party is a state-owned museum, the nature of its public welfare institutions often makes it limited in authorization methods, scope, personnel incentives, and so on, so it can easily lead to insufficient responses to market demand and changes.

2 Proxy authorization model of museum digitized collections

The proxy authorization model refers to the model in which the museum does not directly act as the authorized subject, but entrusts an agent or an authorization platform as an intermediary, authorizes through a contract with the authorized party, and finally uses the digital resources of the collection in the manner agreed to in the contract. In this model, there will be two authorization behaviors: The first time is the authorization by the museum to the agent or the authorization platform, and the second time is the authorization by the agent or the authorization platform to the third party. The process of this type of authorization mode is shown in Figure 2. The Louvre Museum in France and the Solomon R Guggenheim Museum in the United States are typical representatives of this authorization model.

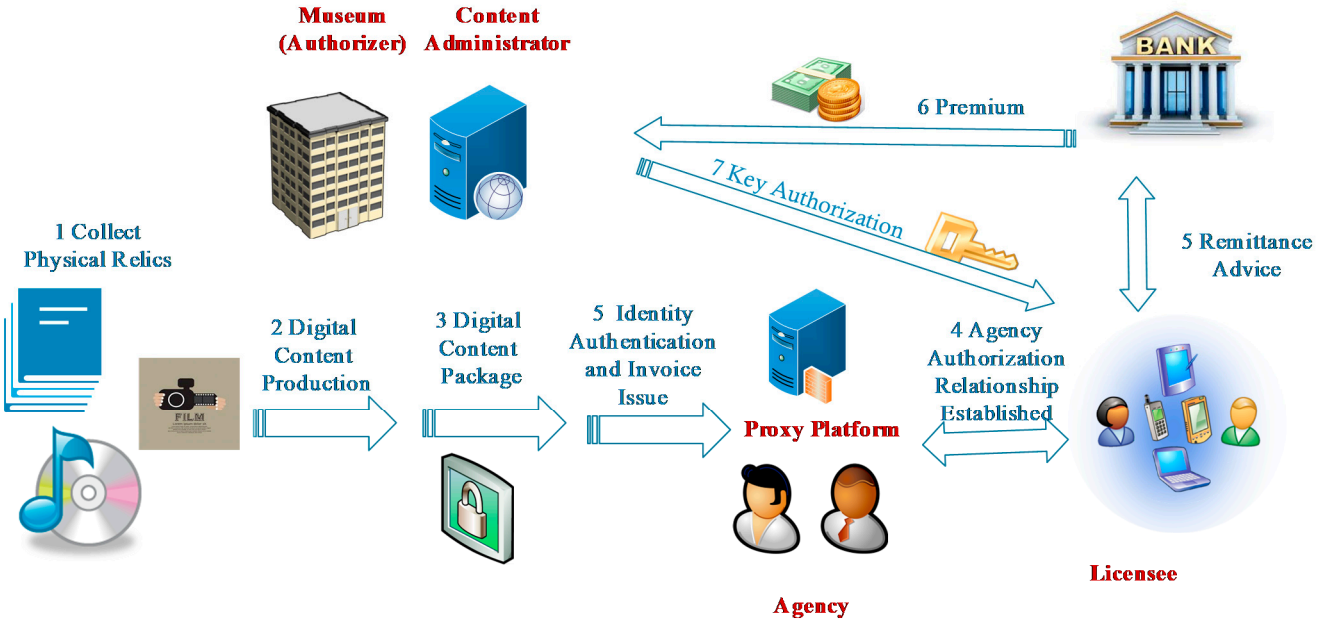

Figure 2. The digital direct authorization model of museum collections.

The entrusted authorization model means that the museum authorizes an agent to sign an authorization contract with the authorized person on behalf of the museum, a common museum proxy authorization model. In the proxy authorization model, agents as authorized intermediaries often have rich authorization management experience and mature customer groups, respond quickly to market demand, and have strong marketing capabilities, which can assist museums in rapidly opening up the authorization market, thereby promoting museums. The cultural and creative production industry has developed rapidly. However, agents, as market entities dominated by economic interests, tend to ignore the public welfare contained in cultural relics, significantly weakening the museum's ability to control the use of the digital collection by authorized third parties. In this process, third parties are based on market interests driving the development and utilization of authorized resources, so the cultural and economic risks faced by museums will increase accordingly.

The platform authorization model is similar to the entrusted authorization model, but there are differences in the scale of the authorizing party and the authorized party. Under the entrusted authorization model, it is usually one-to-one, that is, a museum entrusts

a company to externally authorize, while under the platform authorization model, it is usually many-to-many, that is, multiple museums, middlemen, and authorized parties concentrated in a certain platform carry out authorization. The platform authorization model not only solves the problem of insufficient hardware facilities when most museums carry out the authorization of digital image resources of cultural relics, but also effectively reduces the transaction cost in the process of authorization of digital cultural relics. However, in the platform authorization model, the digital authorization of collections is mainly carried out in the network environment, which is likely to entail transaction risks, including intellectual property rights infringement.

3 Comprehensive authorization model for museum digital collections

The comprehensive authorization model is a composite authorization model, which is a diversified and differentiated authorization strategy made by the museum based on its actual situation. Possessing a certain brand awareness, a large number of collection images, high social recognition, and a variety of types of authorized objects are necessary conditions for the adoption of a comprehensive authorization model; therefore, it needs to be based on the museum's brand awareness, social influence, collection scale, and organization factors, such as staffing and the type of the subject matter of authorization, in making the relevant decision. The comprehensive authorization model combines the advantages of direct authorization and entrusted authorization and helps to optimize the authorization model of different subjects and maximize value creation. The disadvantage is that the complexity of the comprehensive authorization model increases the transaction cost of the authorization process, which will occupy more museum resources to a certain extent. The Metropolitan Museum of Art adopted a comprehensive authorization model when developing art authorization.

Blockchain is a kind of distributed data storage, which has the characteristics of point-to-point transmission, consensus mechanism, and encryption algorithm. For museums, blockchain technology has great value for the digitization of collections and artworks, especially cultural relics, specimens, and artworks. Blockchain has great potential in the confirmation of digital identities. This technology can generate an ID card based on an encryption algorithm for each institution or each person. It has the characteristics of decentralized data storage, decentralization, and traceability. Making clear value guarantees for each collection can also systematically protect the intellectual property rights of cultural relics and artworks so that the whole process of circulation can be followed. The production of digital content and the mechanism of cryptography comprise the foundation of digital property rights. In recent years, blockchain technology has been used to register and digitize collection-related information and cultural relic owner information, and then record these digital files on the blockchain. Because the blockchain has the characteristics of permanent storage and non-tampering, it can establish a one-to-one correspondence between collections, digital information (including photos, three-dimensional models, etc.), and owners, which can effectively solve cultural relic storage, ownership confirmation, and anti-theft, identification, loss prevention, and other issues.

This research is motivated by the following motivations:

(a) In the 20th century, international museums and governments, based on the mission of preservation and promotion of cultural policies to protect cultural resources, implemented digital plans for various museum collections, so that museums can share digital resources, which will not only help to promote social education, but also benefit the operation of museums.

(b) Under the guidance of the "activation and reproduction" thinking, this research uses a "digital authorization" model for museums to provide online users with information and increase financial resources to become a sustainable development of museum operations.

The main contributions of this work are as follows. This research proposes an authorization mechanism based on blockchain technology for protecting the museum's digital

property rights. The signature and time stamp mechanism of cryptography is used to achieve a non-repudiation mechanism, and the smart contract achieves transparency, unforgeability, and traceability; this mechanism will thereby solving the above-mentioned problems faced by museum-digital rights management.

The rest of this article is organized as follows. The second section provides preliminary knowledge. The third section discusses the proposed methods for two kinds of authority mechanisms in the business model. The fourth section presents an analysis of the proposed scheme. The fifth section includes a discussion and comparison of the proposed scheme with related works. Finally, we present the conclusion and future works.

2. Preliminary

2.1. Smart Contract

A smart contract is a special agreement that is used when making a contract in the blockchain. It contains code functions and can interact with other contracts, guide decisions, store data, etc. The main force of smart contracts is to provide verification and execution of the conditions stipulated in the contract. Smart contracts allow credible transactions without the need for a third party. These transactions are traceable and irreversible. The concept of smart contracts was first proposed in 1994 by Nick Szabo [27,28], a computer scientist and cryptography expert. The purpose of smart contracts is to provide better security than traditional contract methods and to reduce other transaction costs associated with the contract.

2.2. ECDSA

In cryptography, the Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA), which uses elliptic curve cryptography [29]. As with elliptic-curve cryptography in general, the bit size of the public key believed to be needed for ECDSA is about twice the size of the security level, in bits. For example, at a security level of 80 bits (meaning an attacker requires a maximum of about 2^{80} operations to find the private key), the size of an ECDSA public key would be 160 bits, whereas the size of a DSA public key is at least 1024 bits. On the other hand, the signature size is the same for both DSA and ECDSA: Approximately $4t$ bits, where t is the security level measured in bits; that is, about 320 bits for a security level of 80 bits.

The signature and verification process of ECDSA is as follows: Suppose Alice wants to send a message to Bob. Initially, both parties must reach a consensus on the curve parameters (CURVE, G, n). In addition to the field equation of the curve, the base point G on the curve and the multiplication order n of the base point G are also required. Alice also needs a private key, d_A and a public key, Q_A, where $Q_A = d_A G$. If the message Alice wants to send is m, Alice needs to choose a random value k between [1, n − 1]: Calculate $z = h(m)$, $(x_1, y_1) = kG$, $r = x_1 \mathrm{mod} n$, $s = k^{-1}(z + rd_A) \mathrm{mod} n$, and send the ECDSA signature pair (r, s) together with the original message m to Bob. After receiving the signature pair (r, s) and the original message m, Bob will verify the correctness of the ECDSA signature. Bob first calculates $z' = h(m)$, $u_1 = z's^{-1} \mathrm{mod} n$, $u_2 = rs^{-1} \mathrm{mod} n$, $(x_1', y_1') = u_1 G + u_2 Q_A$, $r \stackrel{?}{=} x_1' \mathrm{mod} n$, and if it passes the verification, then Bob confirms that the ECDSA signature and message m sent by Alice are correct.

2.3. Bilinear Pairings

The bilinear map was proposed by Boneh et al. in 2001 [30]. Later, Chen et al. applied this in the medical care field [31,32]. Let G_1 be a cyclic additive group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group with the same order q. Let $e : G_1 * G_1 \rightarrow G_2$ be a map with the following properties:

(a) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, $P, Q \in G_1$, $a, b \in Z_q$.
(b) Non-degeneracy: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$, in other words, the map does not send all pairs in $G_1 * G_1$ to the identity in G_2.
(c) Computability: There is an efficient algorithm to compute $e(P, Q)$, $P, Q \in G_1$.

2.4. Proxy Re-Encryption

In 1998, Blaze et al. [33] proposed atomic proxy cryptography for the first time, in which a semi-trusted proxy computes a function that converts ciphertexts for Alice into ciphertexts for Bob without seeing the underlying plaintext. In Elliptic Curve Based Proxy Re-Encryption, the authors combined elliptic curve, bilinear mapping, and proxy re-encryption and proposed the Elliptic Curve based proxy re-encryption. In their scheme, with setting up a large prime number and G, which is a point on elliptic curve E of order n, the proxy is entrusted with delegation key bG/a to change ciphertext from Alice to Bob via computing $(raGbG/a, rG^2 + P_m)$, where P_m is a point on the elliptic curve that embeds the message m in the elliptic curve equation f (i.e., $P_m = f(m)$).

Then we can calculate the message m by finding inverse as $f^{-1}(P_m)$. The proxy re-encryption is a natural application to secure the file system. The following scenarios are the Elliptic Curve based proxy re-encryption mechanism.

(a) System parameter establishment

Let E be an elliptic curve over a limited field F_q, where q is a large prime number, and G is a point on the elliptic curve E of order n. Let $Z_n{}^*$ be a multiplicative group. Let the elliptic curve equation f denote the message embedding function, which maps the message m to a point P_m on E.

(b) Key generation

Alice randomly selects a positive integer $a \in Z_n{}^*$ as his/her private key and calculates aG as the public key. Bob randomly selects a positive integer $b \in Z_n{}^*$ as the private key and calculates bG as Bob's public key.

(c) Alice encrypts the plaintext m:

1. P_m is the embedding message, which is calculated by $f(m)$: $P_m = f(m)$;
2. generate an arbitrary number $r \in Z_n{}^*$ and output the ciphertext $(C_1, C_2) = (raG, rG^2 + P_m)$;
3. send the ciphertext (C_1, C_2) to the proxy.

(d) Generation of the re-encryption key:

1. Alice wants to authorize the information to Bob such that Bob can decrypt the ciphertext; Alice sends the proxy key $\pi_{A\to B} = bG/a$ to the proxy.
2. The semi-honest agent proxy re-encrypts the ciphertext (C_1, C_2) into $(C_1{}', C_2{}')$ and sends it to Bob.

(e) Re-encryption process:

1. For the ciphertext $(C_1, C_2) = (raG, rG^2 + P_m)$, the proxy uses the re-encryption key to re-encrypt (C_1, C_2) into $(C_1{}', C_2{}')$.
2. $(C_1{}', C_2{}')$
   $=(raG\pi_{A\to B}, rG^2 + P_m)$
   $=(raGbG/a, rG^2 + P_m)$
   $=(rbG^2, rG^2 + P_m)$
3. The proxy sends the converted ciphertext $(C_1{}', C_2{}') = (rbG^2, rG^2 + P_m)$ to Bob.

(f) Bob decrypts the ciphertext:

1. Bob can decrypt the embedding message P_m with key b:$P_m = C_2{}' - b^{-1}C_1{}'$;
2. then apply the inverse of the function f to get the original message m from P_m: $m = f^{-1}(P_m)$.

3. Method

3.1. System Architecture

Figure 3 is the system architecture diagram.

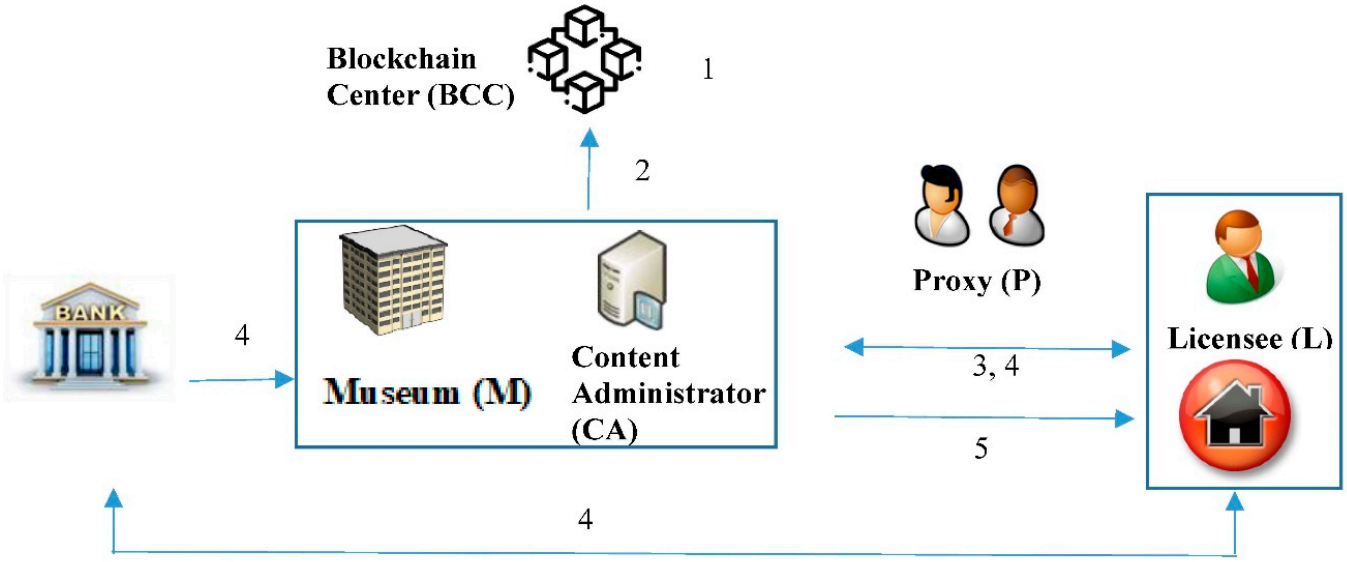

Figure 3. The system architecture.

In this study, we use the Elliptic Curve Digital Signature Algorithm (ECDSA), blockchain, and smart contracts to design a traceable authorization mechanism for the museum's digital content resource. There are six parties involved in this study: Museum (M), Content Administrator (CA), Licensee (L), Blockchain Center (BCC), Proxy (P), and Bank (B).

(a) Museum (M): The museum is the owner of the digital content. The museum collects the cultural relics and is responsible for the generation and management of the museum's digital content resource. The digital content resource is classified and protected by the museum.
(b) Content Administrator (CA): The CA is a cloud platform of the museum. It is responsible for reviewing the Licensee's request to determine 'allow or not' to access the digital content resource.
(c) Licensee (L): When citizens or institutions want to access the digital content resource of the museum, the Licensee should pay a premium to the museum.
(d) Blockchain Center (BCC): This center records the access information of the digital right resource for the Licensee. The BCC accepts the parties' registration and issues the identity certificate and public/private key pair to each party.
(e) Proxy (P): The proxy is an agency of the museum. After CA authenticates the Licensee's identity, P is responsible for actually cloud authorization for the Licensee to access the museum's digital content resource.
(f) Bank (B): Bank is authorized by a Licensee to pay a premium to the museum. We briefly illustrate the scenarios in the following steps.

- Step 1: Registration phase:

Museum, Licensee, Proxy, and Bank need to register with Blockchain Center; the Blockchain Center issues the identity certificate and public/private key pair to each party.

- Step 2: Digital content production phase:

The DCA classifies the museum's resources, encrypts these resources into a protected digital resource, and then stores it in the CA. The CA also uploads the detailed categories into the Blockchain center.

- Step 3: Authentication phase and issuing invoice phase:

After the Licensee proposes to access digital resource requests, the CA reviews the Licensee's qualifications and then issues the invoice.

- Step 4: Payment phase:

After payment, the Licensee requests the Bank to issue a certificate for the museum to authenticate this payment. The Content Administer then authenticates the Licensee's identity. The Content Administer performs one of the following cases.

Case 1: Generates the authorized key to the Licensee directly.

Case 2: Generates a proxy key to the Agency, and the Agency transfers it to the Licensee.

- Step 5: Digital content browsing phase:

After the Licensee receives the authorized key, the Licensee uses it to decrypt the protected digital content. The digital content can be read (or played) normally.

3.2. Smart Contract Initialization

In the proposed architecture, blockchain technology is applied. During the authentication and authorization process, some key information will be saved and verified through the blockchain. The key information in the blockchain is defined in the smart contract. The following is the blockchain smart contract structure for the proposed scheme (Scheme 1).

```
struct smart contract lminf/lainf/aminf{
        string lm/la/am id;
        string lm/la/am detail;
        string lm/la/am cert;
        string lm/la/am tsp;
}
struct smart contract mlinf/mainf/alinf {
        string ml/ma/al id;
        string ml/ma/al detail;
        string ml/ma/al tid;
        string ml/ma/al tsp;
}
```

```
struct smart contract lcinf/lpinf/pcinf{
        string lc/lp/pc id;
        string lc/lp/pc detail;
        string lc/lp/pc payment;
        string lc/lp/pc tsp;
}
struct smart contract clinf/cpinf/plinf {
        string cl/cp/pl id;
        string cl/cp/pl detail;
        string cl/cp/pl key;
        string cl/cp/pl tsp;
}
string keypairs;
string count;
```

Scheme 1. the blockchain smart contract initialization structure.

In the proposed smart contract, we have developed key information that will be stored in the blockchain. In the structure of the lm/la/am smart contract, we developed the field of id (identification), transaction detail, certificate, and timestamp. In the structure of the ml/ma/al smart contract, we developed the field of id, transaction detail, transaction id, and timestamp. In the structure of lc/lp/pc smart contract, we developed the field of id, transaction detail, payment information, and timestamp. In the structure of the cl/cp/pl smart contract, we developed the field of id, transaction detail, authentication key, and timestamp. In the initialization phase, the blockchain center also issues the public and private key pairs for all roles.

3.3. Registration Phase

The Licensee (L), Content Administrator (CA), and Proxy (P) should register with the Blockchain Center (BCC) and obtain a relative public/private key pair. The Licensee (L) and Proxy (P) also get a digital certificate of identity from the Blockchain Center via a secure channel. The system role X can represent the Licensee (L), Content Administrator (CA), and Proxy (P). Figure 4 shows the flowchart of the registration phase.

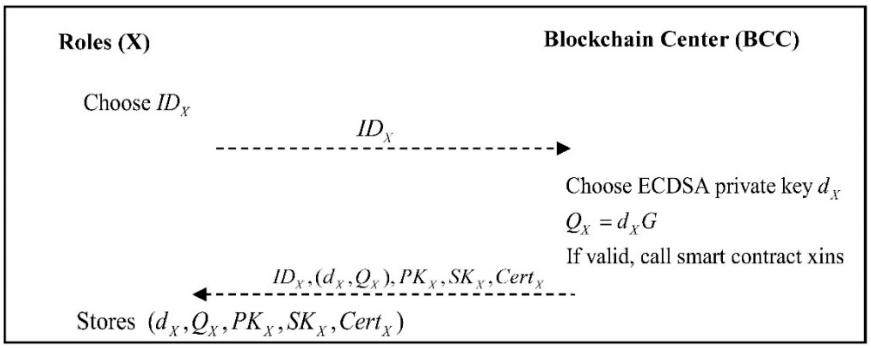

Figure 4. Each role of the system registers with the Blockchain Center.

- Step 1: Role X generates an identity ID_X, and sends it to the Blockchain Center.
- Step 2: The Blockchain center generates an ECDSA private key d_X based on the role X, calculates:

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

If the identity of the registered role is verified, the smart contract Xins will be triggered, and the content is presented as follows (Scheme 2):

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

Scheme 2. The smart contract Xins.

Then the blockchain center will transmit $ID_X, (d_X, Q_X), PK_X, SK_X, Cert_X$ to role X.

- Step 3: The role X stores $(d_X, Q_X, PK_X, SK_X, Cert_X)$.

3.4. Digital Content Production Phase

The museum collects many precious cultural relics. The digital content production process of valuable cultural relics involves a specific process. In general, experts and scholars classify (such as biological classification, antiquities classification, etc.), grade (grade of antiquities is divided into general, important, national treasures, etc.), and clarify the importance (such as rare or era significance or endangered species, etc.), and then different competent authorities proceed with various kinds of appointments. Finally, it is handed over to professional and technical personnel to produce digital content through photography and 3D surroundings.

In this phase, we will focus on illustrating the protection technology of digital content. Figure 3 shows the production flowchart of protected digital content. To enhance performance, we use the digital envelope for implementation. That is, the Content Administrator (CA) uses the symmetry key to encrypt the digital content, and then uses the ElGamal-based system of the public-key system to protect the symmetry key. Figure 5 shows the flowchart of the digital content production phase.

- Step 1: Content Administrator (CA) collects cultural relics in a systematic and planned way according to the categories of different collections. CA also uses information technology to convert the collected media data into a form that can be stored, processed, and edited.
- Step 2: CA encrypts these encoded multimedia data with KeyID and Seed, organizes and categorizes each digitized archive resource, and records the data description of

the archive itself, as an annotation explanation for the archive itself and various media materials, as well as an indexing tool for users to inquire.

- Step 3: Through the overall planning of the collection environment, a suitable information system can be constructed, and the functions of digital data preservation and management can be achieved through the operation of the system. When a Licensee wants to access these multimedia materials, it must first obtain legal authorization from the Content Administrator (CA).
- Step 4: The CA will provide the Licensee with an authorization key; the Licensee can use the authorization key to unlock the information provided by the CA and get a decryption key, which can be used to obtain the plaintext of multimedia messages. The details will be introduced in the following phase.

Figure 5. Digital content production phase.

3.5. Authentication and Issuing Invoice Phase

3.5.1. Case 1: Direct Authorization

After reviewing the Licensee's identity, the Content Administrator generates a transaction ID and invoice to the Licensee. We present the flowchart of the authentication and issuing an invoice phase for direct authorization in Figure 6.

- Step 1: The Licensee generates a random value k_{L-M}, calculates:

$$z_{L-M} = h(ID_L, M_{L-M}, Cert_L, TS_{L-M}, ID_{BC}), \tag{2}$$

$$(x_{L-M}, y_{L-M}) = k_{L-M}G, \tag{3}$$

$$r_{L-M} = x_{L-M} \bmod n, \tag{4}$$

$$s_{L-M} = {k_{L-M}}^{-1}(z_{L-M} + r_{L-M}d_L) \bmod n, \tag{5}$$

$$Enc_{L-M} = E_{PK_M}(ID_L, M_{L-M}, Cert_L, TS_{L-M}, ID_{BC}), \tag{6}$$

and sends $ID_L, Enc_{L-M}, (r_{L-M}, s_{L-M})$ to the content administrator.

The ID_L is encrypted to check integrity. The second ID_L is to show the Licensee's identity to the content administrator.

- Step 2: The Content Administrator first calculates:

$$(ID_L, M_{L-M}, Cert_L, TS_{L-M}, ID_{BC}) = D_{SK_M}(Enc_{L-M}), \tag{7}$$

uses

$$TS_{NOW} - TS_{L-M} \leq \Delta T \tag{8}$$

to confirm whether the timestamp is valid, verifies $Cert_L$ with PK_L, verifies the correctness of the ECDSA signature, then calculates:

$${z_{L-M}}' = h(ID_L, M_{L-M}, Cert_L, TS_{L-M}, ID_{BC}), \tag{9}$$

$$u_{L-M1} = {z_{L-M}}'{s_{L-M}}^{-1} \bmod n, \tag{10}$$

$$u_{L-M2} = r_{L-M}{s_{L-M}}^{-1} \bmod n, \tag{11}$$

$$({x_{L-M}}', {y_{L-M}}') = u_{L-M1}G + u_{L-M2}Q_L, \tag{12}$$

$${x_{L-M}}' \stackrel{?}{=} r_{L-M} \bmod n. \tag{13}$$

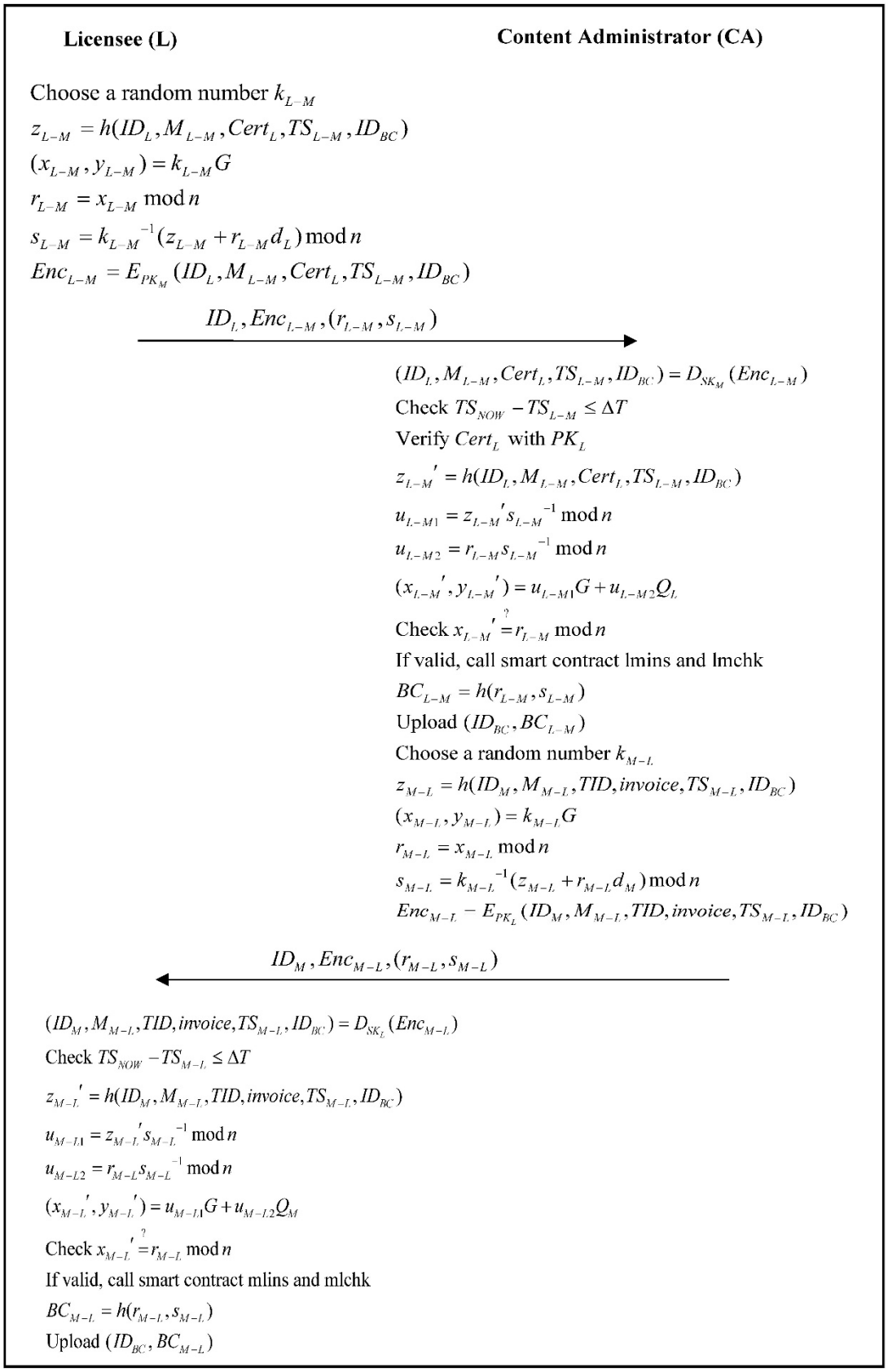

Figure 6. Authentication and issuing invoice phase (direct authorization).

If the verification is passed, CA will get the relevant content request information and trigger the smart contracts lmins and lmchk. The content is as follows (Scheme 3):

```
function insert smart contract lmins(
string lm_id, string lm_detail,
string lm_cert, string lm_tsp) {
        count ++;
        lm[count].id = id;
        lm[count].detail = detail;
        lm[count].cert = cert;
        lm[count].tsp = tsp;
}
sign string l_key (lm_id, lm_detail,
lm_cert, lm_tsp);
```

```
verify string l_key (lm_id, lm_detail,
lm_cert, lm_tsp);
function check smart contract lmchk(
string lm_id, string lm_detail,
string lm_cert, string lm_tsp) {
        return lm_id.exist;
        return lm_detail.exist;
        return lm_cert.exist;
        return lm_tsp.exist;
}
```

Scheme 3. The smart contracts lmins and lmchk.

The CA calculates:

$$BC_{L-M} = h(r_{L-M}, s_{L-M}), \tag{14}$$

(ID_{BC}, BC_{L-M}) will also be uploaded to the blockchain center. Then the CA generates a random value k_{M-L} and calculates:

$$z_{M-L} = h(ID_M, M_{M-L}, TID, invoice, TS_{M-L}, ID_{BC}), \tag{15}$$

$$(x_{M-L}, y_{M-L}) = k_{M-L}G, \tag{16}$$

$$r_{M-L} = x_{M-L} \bmod n, \tag{17}$$

$$s_{M-L} = {k_{M-L}}^{-1}(z_{M-L} + r_{M-L}d_M) \bmod n, \tag{18}$$

$$Enc_{M-L} = E_{PK_L}(ID_M, M_{M-L}, TID, invoice, TS_{M-L}, ID_{BC}), \tag{19}$$

and sends $ID_M, Enc_{M-L}, (r_{M-L}, s_{M-L})$ to the Licensee.

- Step 3: The Licensee first calculates:

$$(ID_M, M_{M-L}, TID, invoice, TS_{M-L}, ID_{BC}) = D_{SK_L}(Enc_{M-L}), \tag{20}$$

uses

$$TS_{NOW} - TS_{M-L} \leq \Delta T \tag{21}$$

to confirm whether the timestamp is valid, verifies the correctness of the ECDSA signature, then calculates:

$${z_{M-L}}' = h(IID_M, M_{M-L}, TID, invoice, TS_{M-L}, ID_{BC}), \tag{22}$$

$$u_{M-L1} = {z_{M-L}}' {s_{M-L}}^{-1} \bmod n, \tag{23}$$

$$u_{M-L2} = r_{M-L} {s_{M-L}}^{-1} \bmod n, \tag{24}$$

$$({x_{M-L}}', {y_{M-L}}') = u_{M-L1}G + u_{M-L2}Q_M, \tag{25}$$

$${x_{M-L}}' \stackrel{?}{=} r_{M-L} \bmod n. \tag{26}$$

If the verification is passed, the content request information is confirmed by CA, and the smart contracts mlins and mlchk will be sent. The content is as follows (Scheme 4):

```
function insert smart contract mlins(
string ml_id, string ml_detail,
string ml_tid, string ml_tsp) {
        count ++;
         ml[count].id = id;
        ml[count].detail = detail;
         ml[count].tid = tid;
         ml[count].tsp = tsp;
}
sign string m_key (ml_id, ml_detail,
ml_tid, ml_tsp);
```

```
verify string m_key (ml_id, ml_detail,
ml_tid, ml_tsp);
function check smart contract mlchk(
string ml_id, string ml_detail,
string ml_tid, string ml_tsp) {
        return ml_id.exist;
       return ml_detail.exist;
        return ml_tid.exist;
        return ml_tsp.exist;
}
```

Scheme 4. The smart contracts mlins and mlchk.

The Licensee calculates:

$$BC_{M-L} = h(r_{M-L}, s_{M-L}), \tag{27}$$

(ID_{BC}, BC_{M-L}) will also be uploaded to the blockchain center.

3.5.2. Case 2: Proxy Authorization

When the Licensee submits an application request to the Proxy, the Proxy transfers it to the CA for verification. After reviewing the Licensee's identity, the CA generates a transaction ID and invoice to the Licensee. We present the flowchart of the authentication and issuing invoice phase (L to P) in Figure 7, the flowchart of the authentication and issuing invoice phase (P to CA) in Figure 8, the flowchart of the authentication and issuing invoice phase (CA to P) in Figure 9, and the flowchart of the authentication and issuing invoice phase (P to L) in Figure 10.

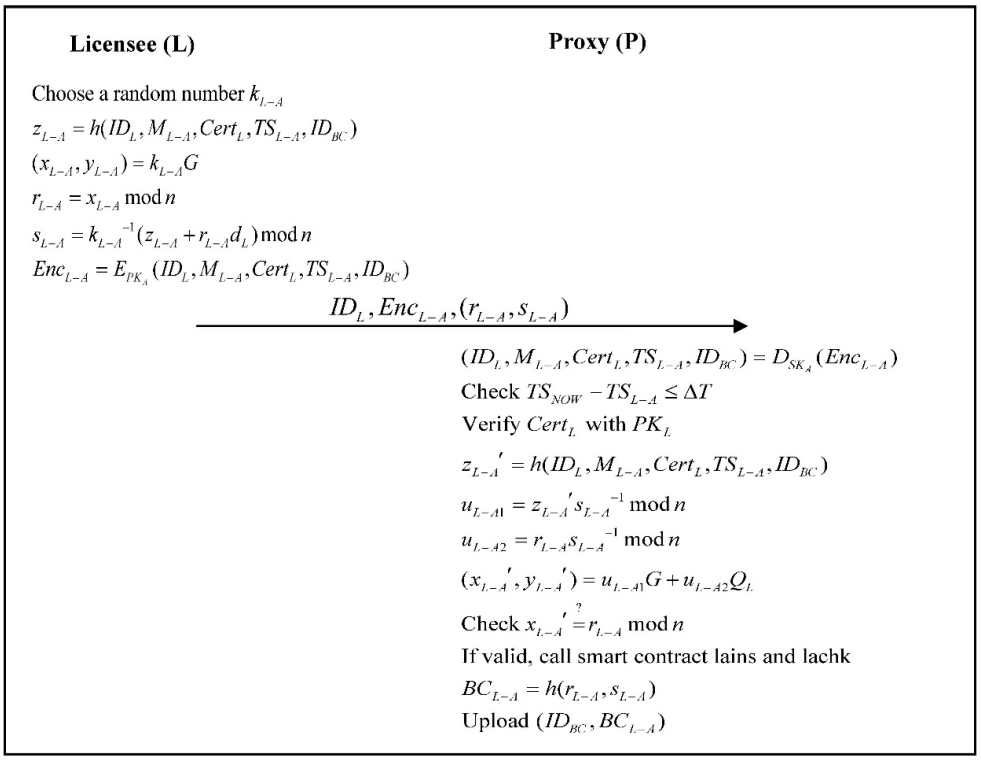

Figure 7. Authentication and issuing invoice phase (L to P).

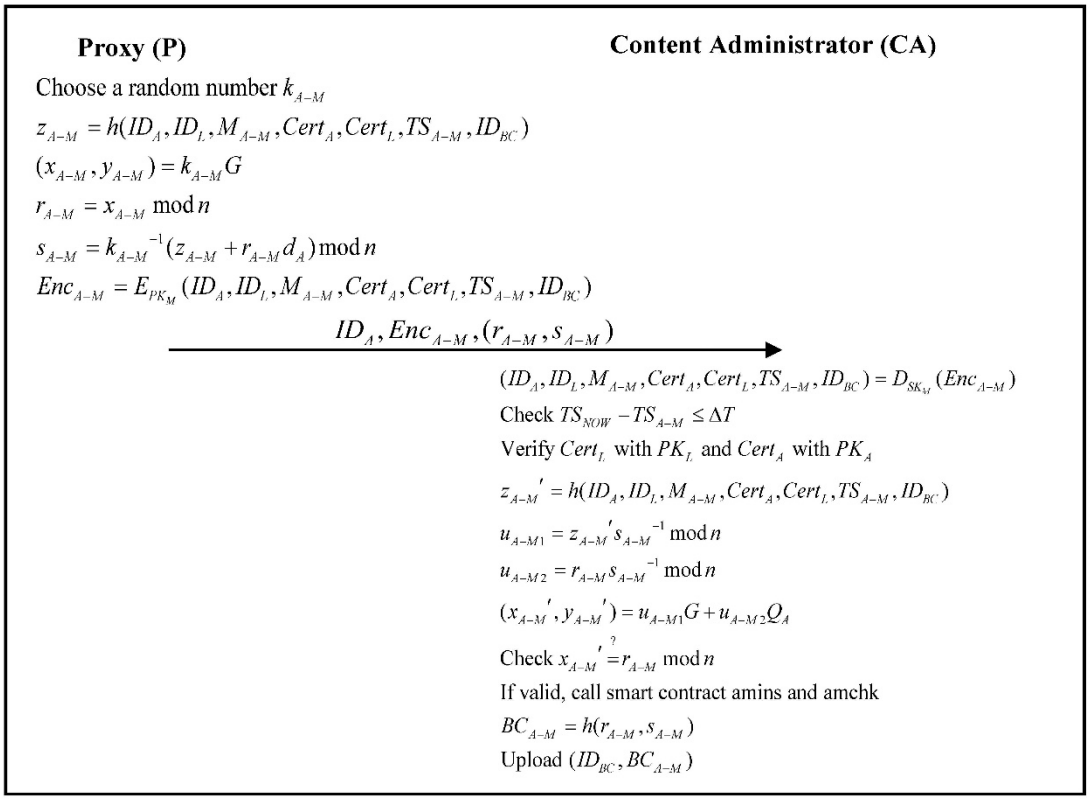

Figure 8. Authentication and issuing invoice phase (P to CA).

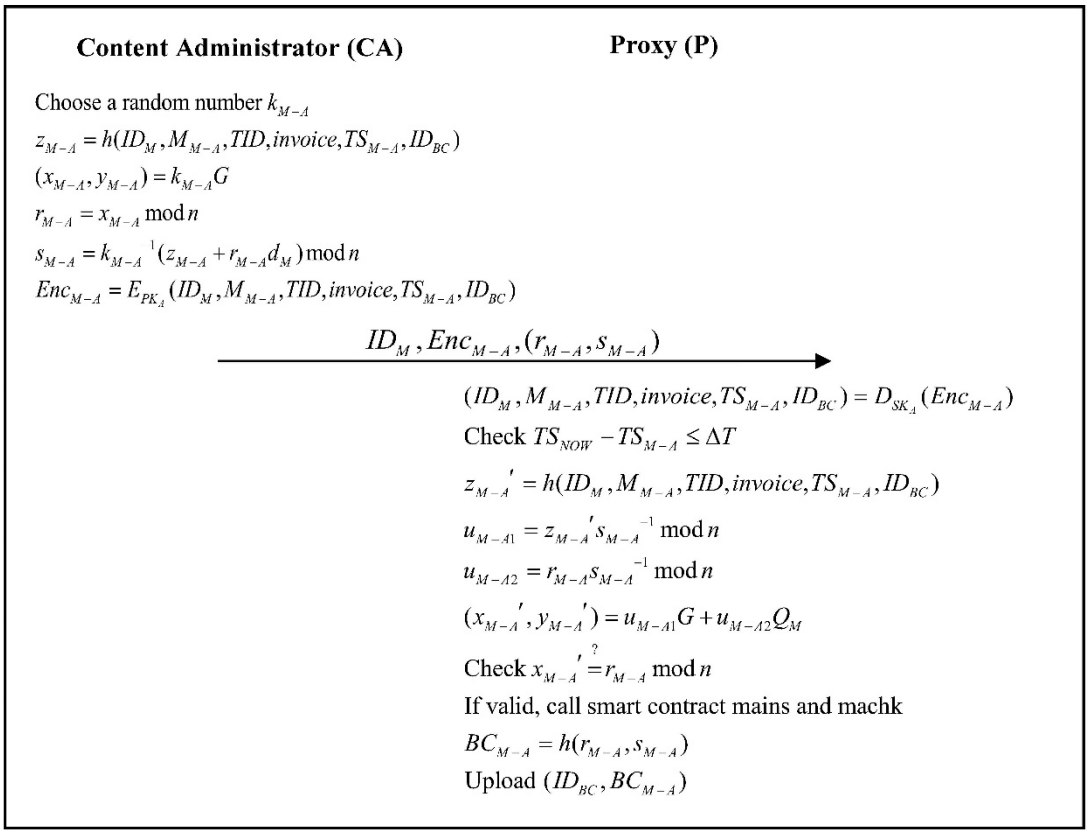

Figure 9. Authentication and issuing invoice phase (CA to P).

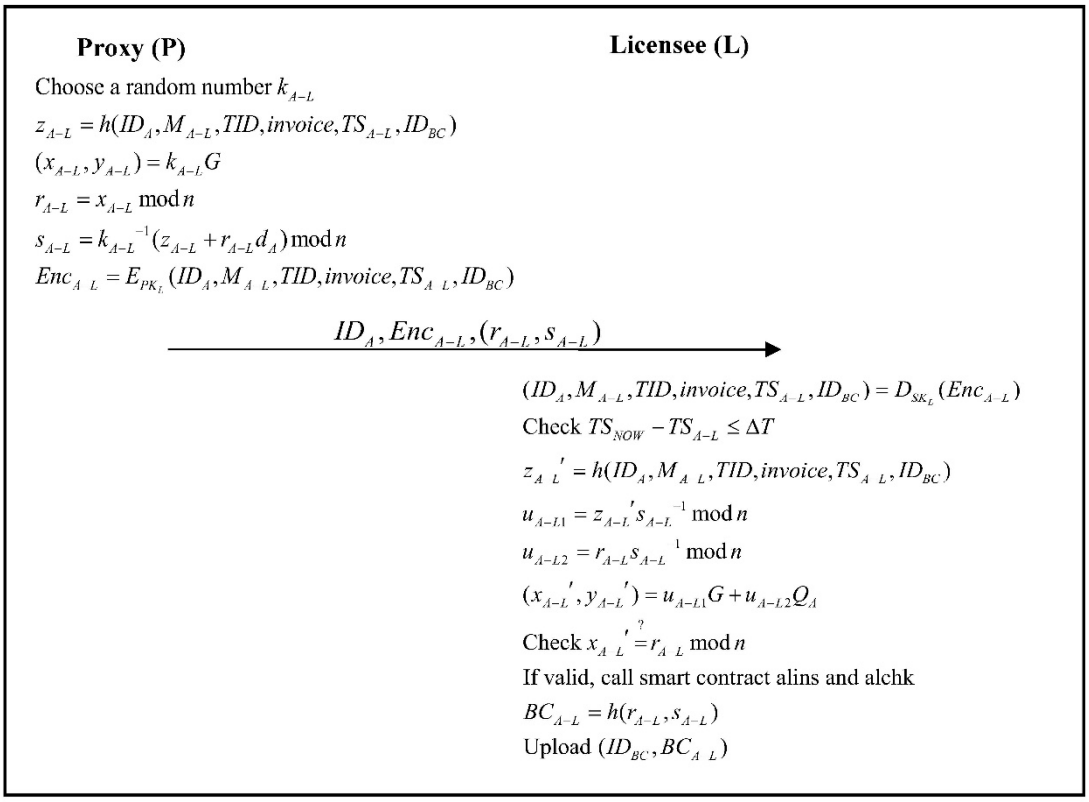

Figure 10. Authentication and issuing invoice phase (P to L).

- Step 1: The Licensee generates a random value k_{L-A}, calculates:

$$z_{L-A} = h(ID_L, M_{L-A}, Cert_L, TS_{L-A}, ID_{BC}), \tag{28}$$

$$(x_{L-A}, y_{L-A}) = k_{L-A}G, \tag{29}$$

$$r_{L-A} = x_{L-A} \bmod n, \tag{30}$$

$$s_{L-A} = k_{L-A}{}^{-1}(z_{L-A} + r_{L-A}d_L) \bmod n, \tag{31}$$

$$Enc_{L-A} = E_{PK_A}(ID_L, M_{L-A}, Cert_L, TS_{L-A}, ID_{BC}), \tag{32}$$

and sends $ID_L, Enc_{L-A}, (r_{L-A}, s_{L-A})$ to the proxy.

- Step 2: The proxy first calculates:

$$(ID_L, M_{L-A}, Cert_L, TS_{L-A}, ID_{BC}) = D_{SK_A}(Enc_{L-A}), \tag{33}$$

uses

$$TS_{NOW} - TS_{L-A} \le \Delta T \tag{34}$$

to confirm whether the timestamp is valid, verifies $Cert_L$ with PK_L, verifies the correctness of the ECDSA signature, and then calculates:

$$z_{L-A}{}' = h(ID_L, M_{L-A}, Cert_L, TS_{L-A}, ID_{BC}), \tag{35}$$

$$u_{L-A1} = z_{L-A}{}' s_{L-A}{}^{-1} \bmod n, \tag{36}$$

$$u_{L-A2} = r_{L-A} s_{L-A}{}^{-1} \bmod n, \tag{37}$$

$$(x_{L-A}{}', y_{L-A}{}') = u_{L-A1}G + u_{L-A2}Q_L, \tag{38}$$

$$x_{L-A}{}' \stackrel{?}{=} r_{L-A} \bmod n. \tag{39}$$

If the verification is passed, the proxy will get the relevant content request information and trigger the smart contracts lains and lachk. The content is as follows (Scheme 5):

```
function insert smart contract lains(
string la_id, string la_detail,
string la_cert, string la_tsp) {
        count ++;
        la[count].id = id;
        la[count].detail = detail;
        la[count].cert = cert;
        la[count].tsp = tsp;
}
sign string l_key (la_id, la_detail,
la_cert, la_tsp);
```

```
verify string l_key (la_id, la_detail,
la_cert, la_tsp);
function check smart contract lachk(
string la_id, string la_detail,
string la_cert, string la_tsp) {
        return la_id.exist;
       return la_detail.exist;
        return la_cert.exist;
        return la_tsp.exist;
}
```

Scheme 5. The smart contracts lains and lachk.

The proxy calculates:

$$BC_{L-A} = h(r_{L-A}, s_{L-A}), \tag{40}$$

(ID_{BC}, BC_{L-A}) will also be uploaded to the blockchain center.

- Step 3: The proxy generates a random value k_{A-M} and calculates:

$$z_{A-M} = h(ID_A, ID_L, M_{A-M}, Cert_A, Cert_L, TS_{A-M}, ID_{BC}), \tag{41}$$

$$(x_{A-M}, y_{A-M}) = k_{A-M}G, \tag{42}$$

$$r_{A-M} = x_{A-M} \bmod n, \tag{43}$$

$$s_{A-M} = {k_{A-M}}^{-1}(z_{A-M} + r_{A-M}d_A) \bmod n, \tag{44}$$

$$Enc_{A-M} = E_{PK_M}(ID_A, ID_L, M_{A-M}, Cert_A, Cert_L, TS_{A-M}, ID_{BC}), \tag{45}$$

and sends $ID_A, Enc_{A-M}, (r_{A-M}, s_{A-M})$ to the CA.

- Step 4: The CA first calculates:

$$(ID_A, ID_L, M_{A-M}, Cert_A, Cert_L, TS_{A-M}, ID_{BC}) = D_{SK_M}(Enc_{A-M}), \tag{46}$$

uses

$$TS_{NOW} - TS_{A-M} \leq \Delta T \tag{47}$$

to confirm whether the timestamp is valid, verifies $Cert_L$ with PK_L and $Cert_A$ with PK_A, verifies the correctness of the ECDSA signature, and then calculates:

$${z_{A-M}}' = h(ID_A, ID_L, M_{A-M}, Cert_A, Cert_L, TS_{A-M}, ID_{BC}), \tag{48}$$

$$u_{A-M1} = {z_{A-M}}' {s_{A-M}}^{-1} \bmod n, \tag{49}$$

$$u_{A-M2} = r_{A-M} {s_{A-M}}^{-1} \bmod n, \tag{50}$$

$$({x_{A-M}}', {y_{A-M}}') = u_{A-M1}G + u_{A-M2}Q_A, \tag{51}$$

$${x_{A-M}}' \stackrel{?}{=} r_{A-M} \bmod n. \tag{52}$$

If the verification is passed, the CA will get the relevant content request information and trigger the smart contracts amins and amchk. The content is as follows (Scheme 6):

```
function insert smart contract amins(
string am_id, string am_detail,
string am_cert, string am_tsp) {
        count ++;
         am[count].id = id;
        am[count].detail = detail;
         am[count].cert = cert;
         am[count].tsp = tsp;
}
sign string a_key (am_id, am_detail,
am_cert, am_tsp);
```

```
verify string a_key (am_id, am_detail,
am_cert, am_tsp);
function check smart contract amchk(
string am_id, string am_detail,
string am_cert, string am_tsp) {
         return am_id.exist;
        return am_detail.exist;
         return am_cert.exist;
         return am_tsp.exist;
}
```

Scheme 6. The smart contracts amins and amchk.

The CA calculates:

$$BC_{A-M} = h(r_{A-M}, s_{A-M}), \tag{53}$$

(ID_{BC}, BC_{A-M}) will also be uploaded to the blockchain center.

- Step 5: The CA generates a random value k_{M-A} and calculates:

$$z_{M-A} = h(ID_M, M_{M-A}, TID, invoice, TS_{M-A}, ID_{BC}), \tag{54}$$

$$(x_{M-A}, y_{M-A}) = k_{M-A}G, \tag{55}$$

$$r_{M-A} = x_{M-A} \bmod n, \tag{56}$$

$$s_{M-A} = {k_{M-A}}^{-1}(z_{M-A} + r_{M-A}d_M) \bmod n, \tag{57}$$

$$Enc_{M-A} = E_{PK_A}(ID_M, M_{M-A}, TID, invoice, TS_{M-A}, ID_{BC}), \tag{58}$$

and sends $ID_M, Enc_{M-A}, (r_{M-A}, s_{M-A})$ to the proxy.

- Step 6: The proxy first calculates:

$$(ID_M, M_{M-A}, TID, invoice, TS_{M-A}, ID_{BC}) = D_{SK_A}(Enc_{M-A}), \tag{59}$$

uses

$$TS_{NOW} - TS_{M-A} \leq \Delta T \tag{60}$$

to confirm whether the timestamp is valid, verifies the correctness of the ECDSA signature, and then calculates:

$${z_{M-A}}' = h(ID_M, M_{M-A}, TID, invoice, TS_{M-A}, ID_{BC}), \tag{61}$$

$$u_{M-A1} = {z_{M-A}}' {s_{M-A}}^{-1} \bmod n, \tag{62}$$

$$u_{M-A2} = r_{M-A} {s_{M-A}}^{-1} \bmod n, \tag{63}$$

$$({x_{M-A}}', {y_{M-A}}') = u_{M-A1}G + u_{M-A2}Q_M, \tag{64}$$

$${x_{M-A}}' \stackrel{?}{=} r_{M-A} \bmod n. \tag{65}$$

If the verification is passed, the content request information is confirmed by the proxy, and the smart contracts mains and machk will be sent. The content is as follows (Scheme 7):

```
function insert smart contract mains(
string ma_id, string ma_detail,
string ma_tid, string ma_tsp) {
        count ++;
         ma[count].id = id;
        ma[count].detail = detail;
         ma[count].tid = tid;
         ma[count].tsp = tsp;
}
sign string m_key (ma_id, ma_detail,
ma_tid, ma_tsp);
```

```
verify string m_key (ma_id, ma_detail,
ma_tid, ma_tsp);
function check smart contract machk(
string ma_id, string ma_detail,
string ma_tid, string ma_tsp) {
        return ma_id.exist;
       return ma_detail.exist;
        return ma_tid.exist;
        return ma_tsp.exist;
}
```

Scheme 7. The smart contracts mains and machk.

The proxy calculates:

$$BC_{M-A} = h(r_{M-A}, s_{M-A}), \tag{66}$$

(ID_{BC}, BC_{M-A}) will also be uploaded to the blockchain center.

- Step 7: The proxy generates a random value k_{A-L} and calculates:

$$z_{A-L} = h(ID_A, M_{A-L}, TID, invoice, TS_{A-L}, ID_{BC}), \tag{67}$$

$$(x_{A-L}, y_{A-L}) = k_{A-L}G, \tag{68}$$

$$r_{A-L} = x_{A-L} \bmod n, \tag{69}$$

$$s_{A-L} = {k_{A-L}}^{-1}(z_{A-L} + r_{A-L}d_A) \bmod n, \tag{70}$$

$$Enc_{A-L} = E_{PK_L}(ID_A, M_{A-L}, TID, invoice, TS_{A-L}, ID_{BC}), \tag{71}$$

and sends $ID_A, Enc_{A-L}, (r_{A-L}, s_{A-L})$ to the Licensee.

- Step 8: The Licensee first calculates:

$$(ID_A, M_{A-L}, TID, invoice, TS_{A-L}, ID_{BC}) = D_{SK_L}(Enc_{A-L}), \tag{72}$$

uses

$$TS_{NOW} - TS_{A-L} \leq \Delta T \tag{73}$$

to confirm whether the timestamp is valid, verifies the correctness of the ECDSA signature, and then calculates:

$${z_{A-L}}' = h(ID_A, M_{A-L}, TID, invoice, TS_{A-L}, ID_{BC}), \tag{74}$$

$$u_{A-L1} = {z_{A-L}}' {s_{A-L}}^{-1} \bmod n, \tag{75}$$

$$u_{A-L2} = r_{A-L} {s_{A-L}}^{-1} \bmod n, \tag{76}$$

$$({x_{A-L}}', {y_{A-L}}') = u_{A-L1}G + u_{A-L2}Q_A, \tag{77}$$

$${x_{A-L}}' \stackrel{?}{=} r_{A-L} \bmod n. \tag{78}$$

If the verification is passed, the content request information is confirmed by the CA, and the smart contracts alins and alchk will be sent. The content is as follows (Scheme 8):

```
function insert smart contract alins(
string al_id, string al_detail,
string al_tid, string al_tsp) {
        count ++;
         al[count].id = id;
        al[count].detail = detail;
         al[count].tid = tid;
         al[count].tsp = tsp;
}
sign string a_key (al_id, al_detail,
al_tid, al_tsp);
```

```
verify string a_key (al_id, al_detail,
al_tid, al_tsp);
function check smart contract alchk(
string al_id, string al_detail,
string al_tid, string al_tsp) {
        return al_id.exist;
       return al_detail.exist;
        return al_tid.exist;
        return al_tsp.exist;
}
```

Scheme 8. The smart contracts alins and alchk.

The Licensee calculates:

$$BC_{A-L} = h(r_{A-L}, s_{A-L}), \tag{79}$$

(ID_{BC}, BC_{A-L}) will also be uploaded to the blockchain center.

3.6. Payment Verification and Browsing Phase

3.6.1. Case 1: Direct Authorization

After the Licensee is paid, the bank must sign and issue the payment certificate to verify. The CA then authenticates the Licensee's identity and bank payment certificate. After that, the CA generates the authorized key, making time-sensitive tokens. After authorization, the Licensee's application (reader or player) can use the authorized key to automatically decrypt the symmetry key. The APP can browse digital content normally. We present the flowchart of the payment verification and browsing phase for direct authorization in Figure 11.

- Step 1: The Licensee generates a random value k_{L-C}, calculates:

$$z_{L-C} = h(ID_L, M_{L-C}, Cert_L, TID, Cert_{pay}, TS_{L-C}, ID_{BC}), \tag{80}$$

$$(x_{L-C}, y_{L-C}) = k_{L-C}G, \tag{81}$$

$$r_{L-C} = x_{L-C} \bmod n, \tag{82}$$

$$s_{L-C} = {k_{L-C}}^{-1}(z_{L-C} + r_{L-C}d_L) \bmod n, \tag{83}$$

$$Enc_{L-C} = E_{PK_C}(ID_L, M_{L-C}, Cert_L, TID, Cert_{pay}, TS_{L-C}, ID_{BC}), \tag{84}$$

and sends $ID_L, Enc_{L-C}, (r_{L-C}, s_{L-C})$ to the content administrator.

- Step 2: The CA first calculates:

$$(ID_L, M_{L-C}, Cert_L, TID, Cert_{pay}, TS_{L-C}, ID_{BC}) = D_{SK_C}(Enc_{L-C}), \tag{85}$$

uses

$$TS_{NOW} - TS_{L-C} \leq \Delta T \tag{86}$$

to confirm whether the timestamp is valid, it verifies $Cert_L$ with PK_L and $Cert_{pay}$ with PK_{BANK}, verifies the correctness of the ECDSA signature, and then calculates:

$${z_{L-C}}' = h(ID_L, M_{L-C}, Cert_L, TID, Cert_{pay}, TS_{L-C}, ID_{BC}), \tag{87}$$

$$u_{L-C1} = {z_{L-C}}'{s_{L-C}}^{-1} \bmod n, \tag{88}$$

$$u_{L-C2} = r_{L-C}{s_{L-C}}^{-1} \bmod n, \tag{89}$$

$$(x_{L-C}', y_{L-C}') = u_{L-C1}G + u_{L-C2}Q_L, \tag{90}$$

$$x_{L-C}' \stackrel{?}{=} r_{L-C} \bmod n. \tag{91}$$

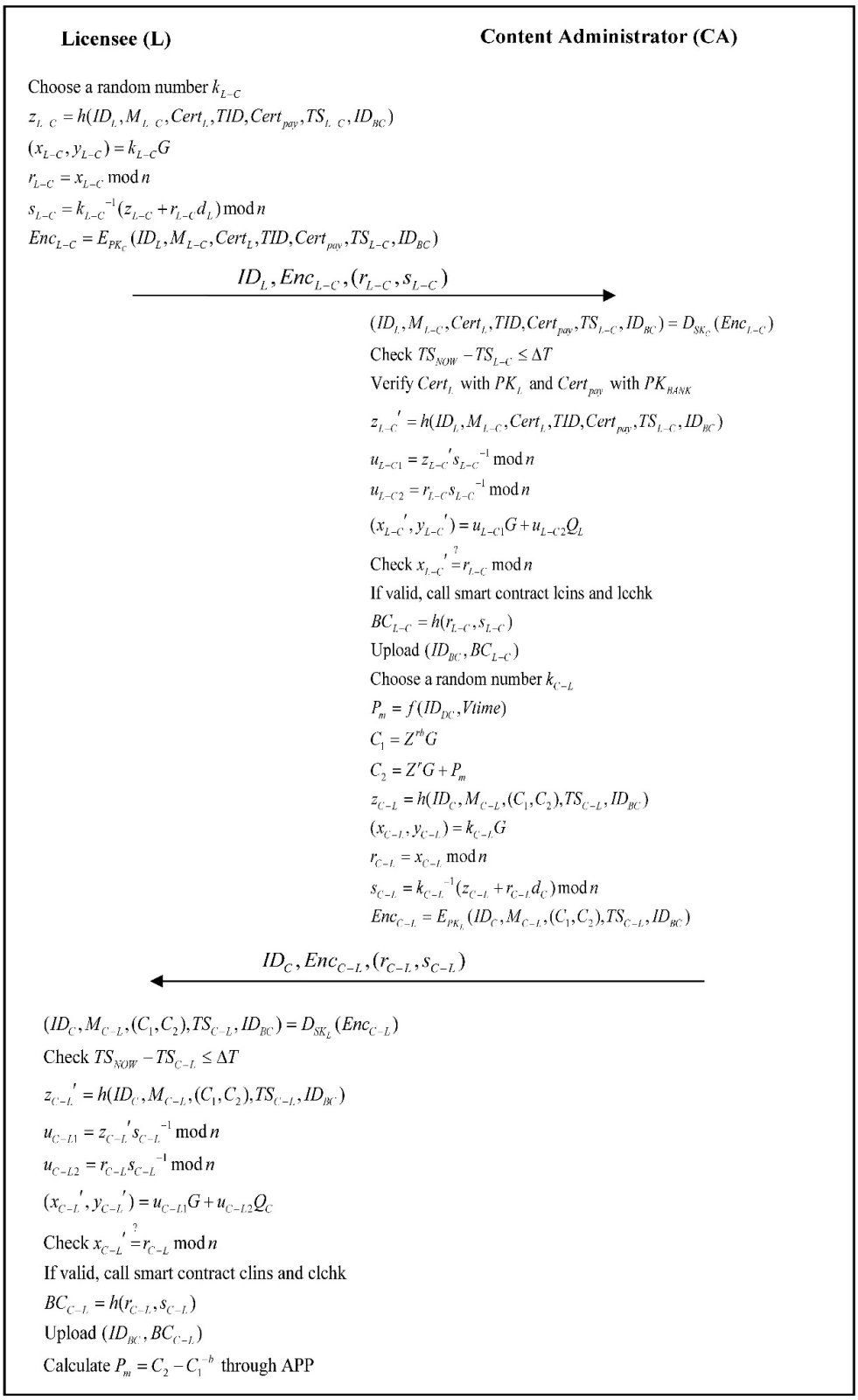

Figure 11. Payment verification and browsing phase (direct authorization).

If the verification is passed, the content administrator will get the relevant payment information and trigger the smart contracts lcins and lcchk. The content is as follows (Scheme 9):

```
function insert smart contract lcins(
string lc_id, string lc_detail,
string lc_payment, string lc_tsp) {
        count ++;
         lc[count].id = id;
        lc[count].detail = detail;
         lc[count].payment = payment;
         lc[count].tsp = tsp;
}
sign string l_key (lc_id, lc_detail,
lc_payment, lc_tsp);
```

```
verify string l_key (lc_id, lc_detail,
lc_payment, lc_tsp);
function check smart contract lcchk(
string lc_id, string lc_detail,
string lc_payment, string lc_tsp) {
         return lc_id.exist;
        return lc_detail.exist;
         return lc_payment.exist;
         return lc_tsp.exist;
}
```

Scheme 9. The smart contracts lcins and lcchk.

The content administrator calculates:

$$BC_{L-C} = h(r_{L-C}, s_{L-C}), \tag{92}$$

(ID_{BC}, BC_{L-C}) will also be uploaded to the blockchain center. Then the content administrator generates a random value k_{C-L} and calculates:

$$P_m = f(ID_{DC}, Vtime), \tag{93}$$

$$C_1 = Z^{rb}G, \tag{94}$$

$$C_2 = Z^{r}G + P_m, \tag{95}$$

$$z_{C-L} = h(ID_C, M_{C-L}, (C_1, C_2), TS_{C-L}, ID_{BC}), \tag{96}$$

$$(x_{C-L}, y_{C-L}) = k_{C-L}G, \tag{97}$$

$$r_{C-L} = x_{C-L} \mathrm{mod} n, \tag{98}$$

$$s_{C-L} = {k_{C-L}}^{-1}(z_{C-L} + r_{C-L}d_C) \mathrm{mod} n, \tag{99}$$

$$Enc_{C-L} = E_{PK_L}(ID_C, M_{C-L}, (C_1, C_2), TS_{C-L}, ID_{BC}), \tag{100}$$

and sends $ID_C, Enc_{C-L}, (r_{C-L}, s_{C-L})$ to the Licensee.

- Step 3: The Licensee first calculates:

$$(ID_C, M_{C-L}, (C_1, C_2), TS_{C-L}, ID_{BC}) = D_{SK_L}(Enc_{C-L}), \tag{101}$$

uses

$$TS_{NOW} - TS_{C-L} \leq \Delta T \tag{102}$$

to confirm whether the timestamp is valid, verifies the correctness of the ECDSA signature, and then calculates:

$${z_{C-L}}' = h(ID_C, M_{C-L}, (C_1, C_2), TS_{C-L}, ID_{BC}), \tag{103}$$

$$u_{C-L1} = {z_{C-L}}' {s_{C-L}}^{-1} \mathrm{mod} n, \tag{104}$$

$$u_{C-L2} = r_{C-L} {s_{C-L}}^{-1} \mathrm{mod} n, \tag{105}$$

$$(x_{C-L}{}', y_{C-L}{}') = u_{C-L1}G + u_{C-L2}Q_C, \tag{106}$$

$$x_{C-L}{}' \stackrel{?}{=} r_{C-L} \bmod n. \tag{107}$$

If the verification is passed, the payment information is confirmed by the content administrator, and the smart contracts clins and clchk will be sent. The content is as follows (Scheme 10):

```
function insert smart contract clins(
string cl_id, string cl_detail,
string cl_key, string cl_tsp) {
        count ++;
         cl[count].id = id;
        cl[count].detail = detail;
         cl[count].key = key;
         cl[count].tsp = tsp;
}
sign string c_key (cl_id, cl_detail,
cl_key, cl_tsp);
```

```
verify string c_key (cl_id, cl_detail,
cl_key, cl_tsp);
function check smart contract clchk(
string cl_id, string cl_detail,
string cl_key, string cl_tsp) {
         return cl_id.exist;
        return cl_detail.exist;
         return cl_key.exist;
         return cl_tsp.exist;
}
```

Scheme 10. The smart contracts clins and clchk.

The Licensee calculates:

$$BC_{C-L} = h(r_{C-L}, s_{C-L}), \tag{108}$$

(ID_{BC}, BC_{C-L}) will also be uploaded to the blockchain center. Finally, the APP calculates:

$$P_m = C_2 - {C_1}^{-b} \tag{109}$$

to successfully obtain the identity of the digital content. This step is performed automatically by the smart contract, and the Licensee cannot skip the verification process privately.

3.6.2. Case 2: Proxy Authorization

After payment, the bank must sign and issue the payment certificate to the Licensee. The Licensee submits the payment certificate to the Proxy, and the Proxy transfers it to the Content Administrator for verification. The CA then authenticates the Licensee's identity and bank payment certificate. After that, the CA generates the authorized key, making time-sensitive tokens. After authorization, the Licensee's application (reader or player) can use the authorized key to automatically decrypt the symmetry key. The APP can browse digital content normally. We present the flowchart of the payment verification and browsing phase (L to P) in Figure 12, the flowchart of the payment verification and browsing phase (P to CA) in Figure 13, the flowchart of the payment verification and browsing phase (CA to P) in Figure 14, and the flowchart of the payment verification and browsing phase (P to L) in Figure 15.

- Step 1: The Licensee generates a random value k_{L-P}, calculates:

$$z_{L-P} = h(ID_L, M_{L-P}, Cert_L, TID, Cert_{pay}, TS_{L-P}, ID_{BC}), \tag{110}$$

$$(x_{L-P}, y_{L-P}) = k_{L-P}G, \tag{111}$$

$$r_{L-P} = x_{L-P} \bmod n, \tag{112}$$

$$s_{L-P} = {k_{L-P}}^{-1}(z_{L-P} + r_{L-P}d_L) \bmod n, \tag{113}$$

$$Enc_{L-P} = E_{PK_P}(ID_L, M_{L-P}, Cert_L, TID, Cert_{pay}, TS_{L-P}, ID_{BC}), \quad (114)$$

and sends $ID_L, Enc_{L-P}, (r_{L-P}, s_{L-P})$ to the proxy.

- Step 2: The Proxy first calculates:

$$(ID_L, M_{L-P}, Cert_L, TID, Cert_{pay}, TS_{L-P}, ID_{BC}) = D_{SK_P}(Enc_{L-P}), \quad (115)$$

uses

$$TS_{NOW} - TS_{L-P} \leq \Delta T \quad (116)$$

to confirm whether the timestamp is valid, verifies $Cert_L$ with PK_L and $Cert_{pay}$ with PK_{BANK}, verifies the correctness of the ECDSA signature, and then calculates:

$$z_{L-P}{}' = h(ID_L, M_{L-P}, Cert_L, TID, Cert_{pay}, TS_{L-P}, ID_{BC}), \quad (117)$$

$$u_{L-P1} = z_{L-P}{}' s_{L-P}{}^{-1} \mathrm{mod} n, \quad (118)$$

$$u_{L-P2} = r_{L-P} s_{L-P}{}^{-1} \mathrm{mod} n, \quad (119)$$

$$(x_{L-P}{}', y_{L-P}{}') = u_{L-P1} G + u_{L-P2} Q_L, \quad (120)$$

$$x_{L-P}{}' \stackrel{?}{=} r_{L-P} \mathrm{mod} n. \quad (121)$$

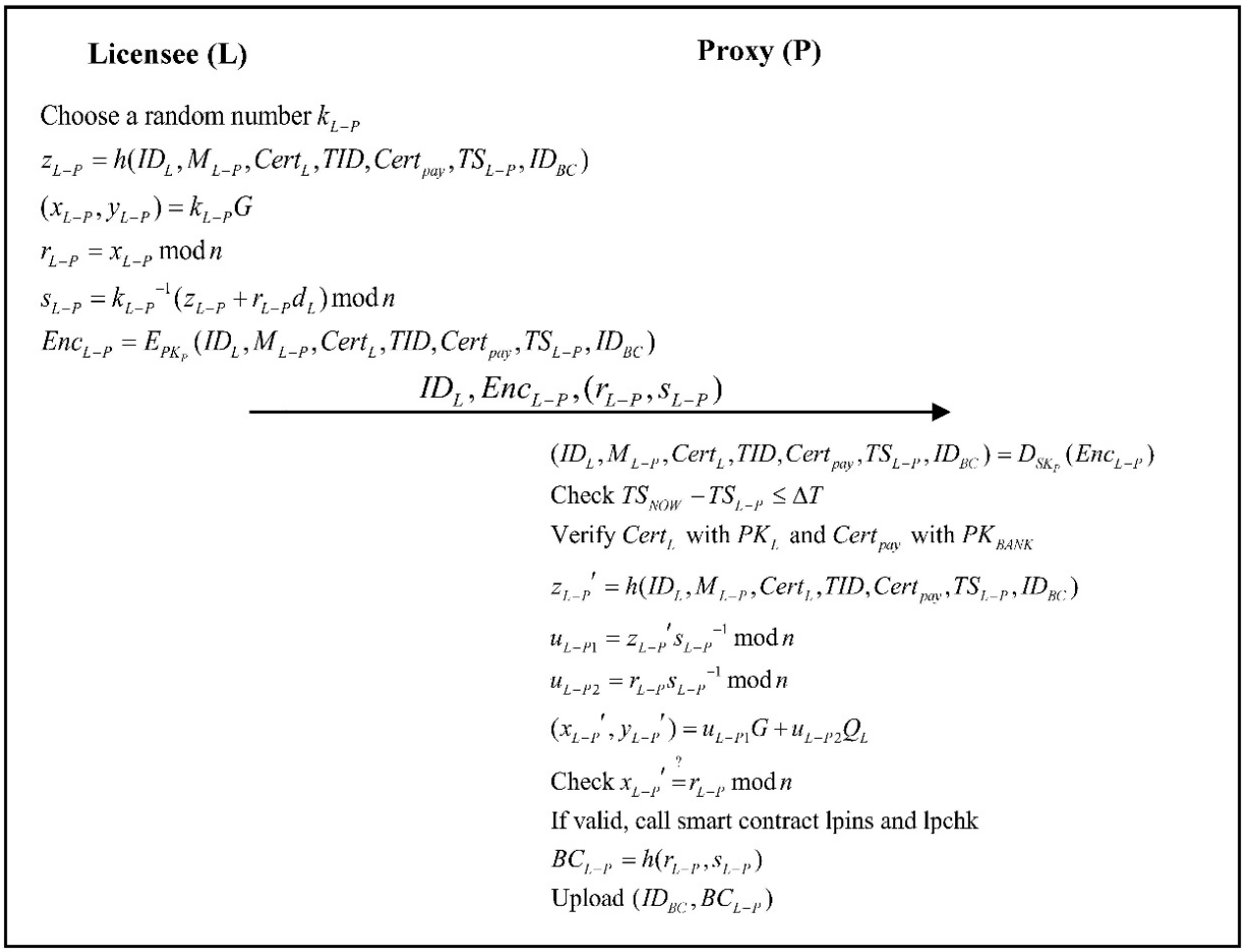

Figure 12. Payment verification and browsing phase (L to P).

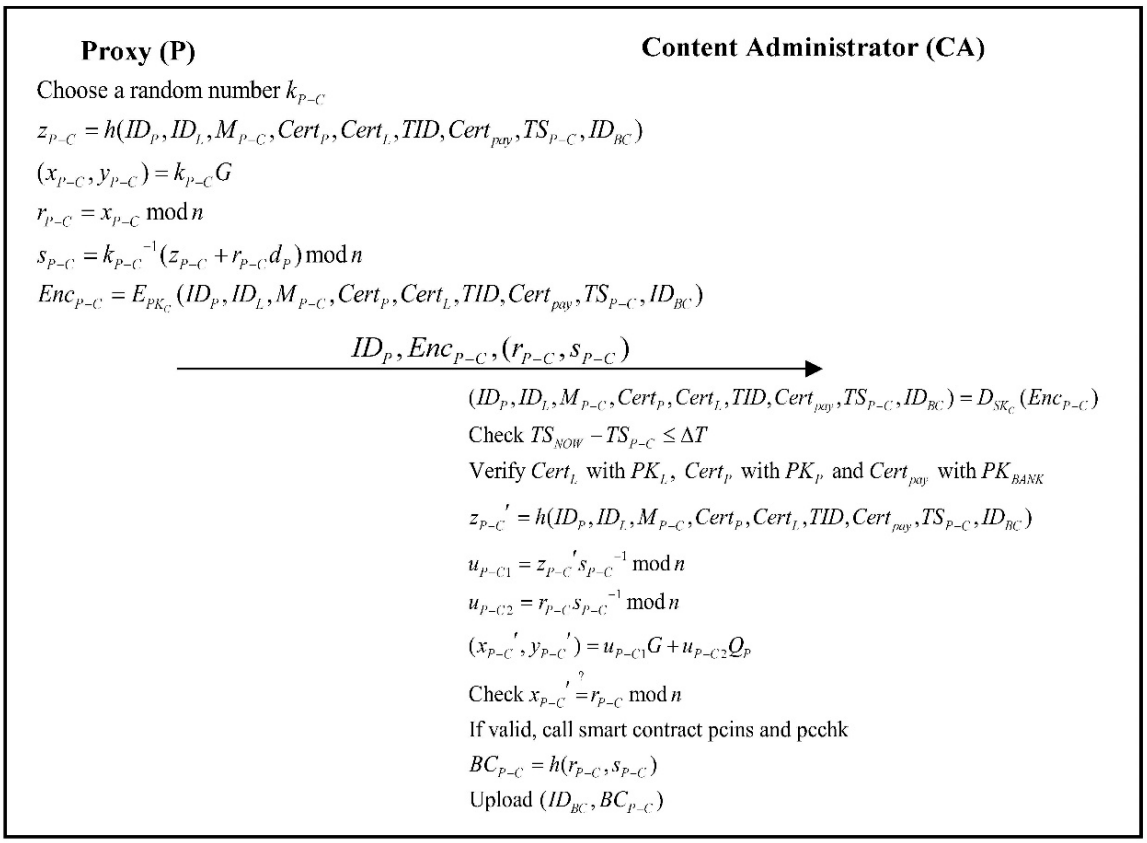

Figure 13. Payment verification and browsing phase (P to CA).

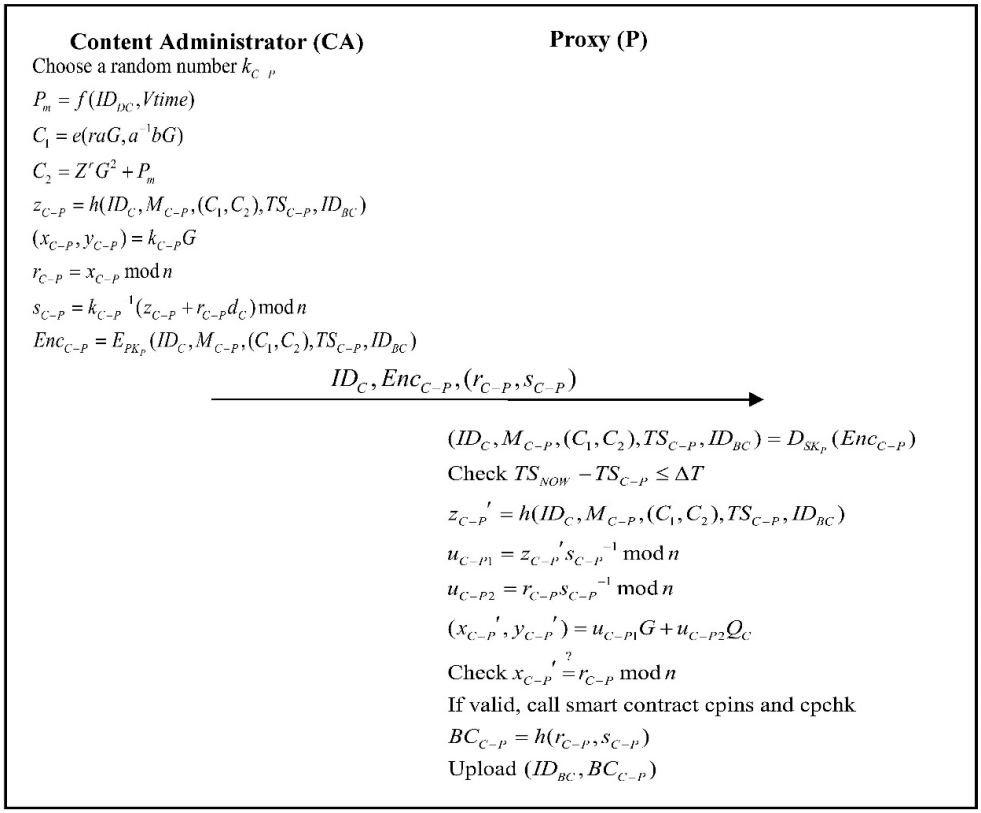

Figure 14. Payment verification and browsing phase (CA to P).

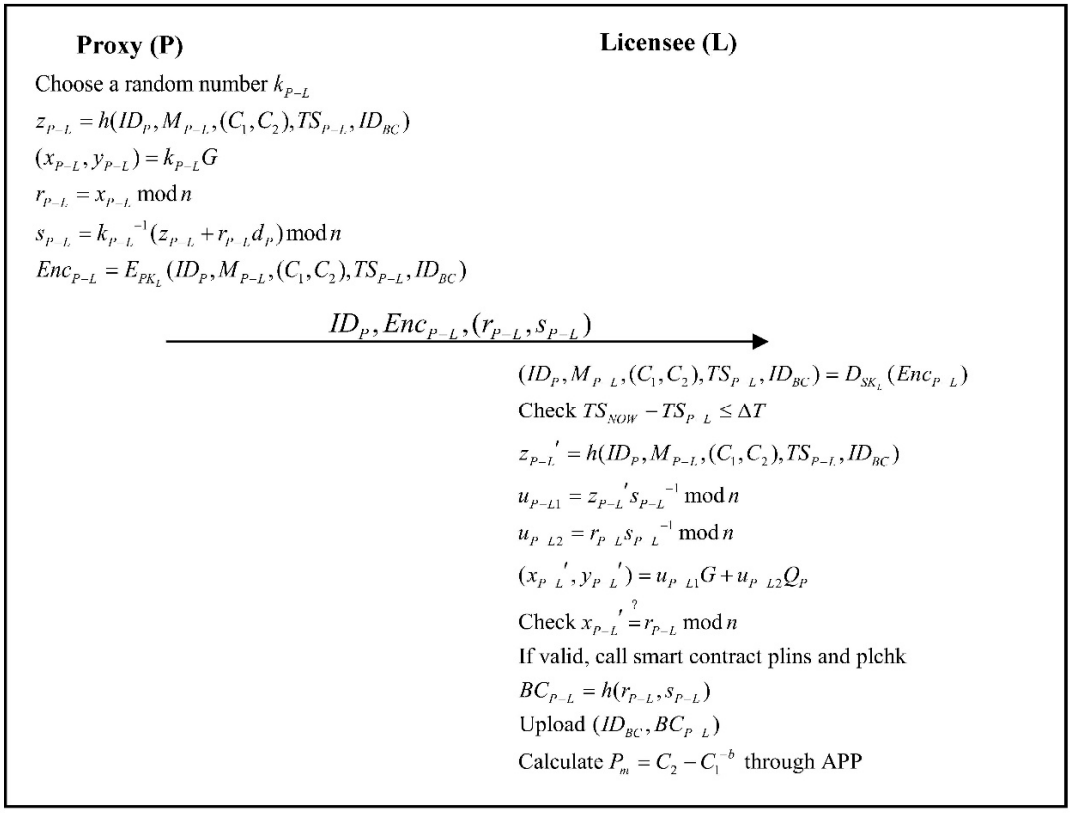

Figure 15. Payment verification and browsing phase (P to L).

If the verification is passed, the proxy will get the relevant payment information and trigger the smart contracts lpins and lpchk. The content is as follows (Scheme 11):

```
function insert smart contract lpins(
string lp_id, string lp_detail,
string lp_payment, string lp_tsp) {
        count ++;
         lp[count].id = id;
        lp[count].detail = detail;
         lp[count].payment = payment;
         lp[count].tsp = tsp;
}
sign string l_key (lp_id, lp_detail,
lp_payment, lp_tsp);
```

```
verify string l_key (lp_id, lp_detail,
lp_payment, lp_tsp);
function check smart contract lpchk(
string lp_id, string lp_detail,
string lp_payment, string lp_tsp) {
         return lp_id.exist;
        return lp_detail.exist;
         return lp_payment.exist;
         return lp_tsp.exist;
}
```

Scheme 11. The smart contracts lpins and lpchk.

The Proxy calculates:

$$BC_{L-P} = h(r_{L-P}, s_{L-P}), \tag{122}$$

(ID_{BC}, BC_{L-P}) will also be uploaded to the blockchain center.

- Step 3: The Proxy then generates a random value k_{P-C} and calculates:

$$z_{P-C} = h(ID_P, ID_L, M_{P-C}, Cert_P, Cert_L, TID, Cert_{pay}, TS_{P-C}, ID_{BC}), \tag{123}$$

$$(x_{P-C}, y_{P-C}) = k_{P-C}G, \tag{124}$$

$$r_{P-C} = x_{P-C} \bmod n, \tag{125}$$

$$s_{P-C} = {k_{P-C}}^{-1}(z_{P-C} + r_{P-C}d_P) \bmod n, \tag{126}$$

$$Enc_{P-C} = E_{PK_C}(ID_P, ID_L, M_{P-C}, Cert_P, Cert_L, TID, Cert_{pay}, TS_{P-C}, ID_{BC}), \tag{127}$$

and sends $ID_P, Enc_{P-C}, (r_{P-C}, s_{P-C})$ to the content administrator.

- Step 4: The CA first calculates:

$$(ID_P, ID_L, M_{P-C}, Cert_P, Cert_L, TID, Cert_{pay}, TS_{P-C}, ID_{BC}) = D_{SK_C}(Enc_{P-C}), \tag{128}$$

uses

$$TS_{NOW} - TS_{P-C} \leq \Delta T \tag{129}$$

to confirm whether the timestamp is valid, verifies $Cert_L$ with PK_L, $Cert_P$ with PK_P and $Cert_{pay}$ with PK_{BANK}, verifies the correctness of the ECDSA signature, and then calculates:

$${z_{P-C}}' = h(ID_P, ID_L, M_{P-C}, Cert_P, Cert_L, TID, Cert_{pay}, TS_{P-C}, ID_{BC}), \tag{130}$$

$$u_{P-C1} = {z_{P-C}}' {s_{P-C}}^{-1} \bmod n, \tag{131}$$

$$u_{P-C2} = r_{P-C} {s_{P-C}}^{-1} \bmod n, \tag{132}$$

$$({x_{P-C}}', {y_{P-C}}') = u_{P-C1}G + u_{P-C2}Q_P, \tag{133}$$

$${x_{P-C}}' \stackrel{?}{=} r_{P-C} \bmod n. \tag{134}$$

If the verification is passed, the content administrator will get the relevant payment information and trigger the smart contracts pcins and pcchk. The content is as follows (Scheme 12):

```
function insert smart contract pcins(
string pc_id, string pc_detail,
string pc_payment, string pc_tsp) {
        count ++;
         pc[count].id = id;
        pc[count].detail = detail;
         pc[count].payment = payment;
         pc[count].tsp = tsp;
}
sign string p_key (pc_id, pc_detail,
pc_cert, pc_tsp);
```

```
verify string p_key (pc_id, pc_detail,
pc_payment, pc_tsp);
function check smart contract pcchk(
string pc_id, string pc_detail,
string pc_payment, string pc_tsp) {
         return pc_id.exist;
        return pc_detail.exist;
         return pc_payment.exist;
         return pc_tsp.exist;
}
```

Scheme 12. The smart contracts pcins and pcchk.

The content administrator calculates:

$$BC_{P-C} = h(r_{P-C}, s_{P-C}), \tag{135}$$

(ID_{BC}, BC_{P-C}) will also be uploaded to the blockchain center.

- Step 5: The content administrator generates a random value k_{C-P} and calculates:

$$P_m = f(ID_{DC}, Vtime), \tag{136}$$

$$C_1 = e(raG, a^{-1}bG), \tag{137}$$

$$C_2 = Z^r G^2 + P_m, \tag{138}$$

$$z_{C-P} = h(ID_C, M_{C-P}, (C_1, C_2), TS_{C-P}, ID_{BC}), \tag{139}$$

$$(x_{C-P}, y_{C-P}) = k_{C-P}G, \tag{140}$$

$$r_{C-P} = x_{C-P} \bmod n, \tag{141}$$

$$s_{C-P} = {k_{C-P}}^{-1}(z_{C-P} + r_{C-P}d_C) \bmod n, \tag{142}$$

$$Enc_{C-P} = E_{PK_P}(ID_C, M_{C-P}, (C_1, C_2), TS_{C-P}, ID_{BC}), \tag{143}$$

and sends $ID_C, Enc_{C-P}, (r_{C-P}, s_{C-P})$ to the proxy.

- Step 6: The Proxy first calculates:

$$(ID_C, M_{C-P}, (C_1, C_2), TS_{C-P}, ID_{BC}) = D_{SK_P}(Enc_{C-P}), \tag{144}$$

uses

$$TS_{NOW} - TS_{C-P} \leq \Delta T \tag{145}$$

to confirm whether the timestamp is valid, it verifies the correctness of the ECDSA signature, and then calculates:

$${z_{C-P}}' = h(ID_C, M_{C-P}, (C_1, C_2), TS_{C-P}, ID_{BC}), \tag{146}$$

$$u_{C-P1} = {z_{C-P}}' {s_{C-P}}^{-1} \bmod n, \tag{147}$$

$$u_{C-P2} = r_{C-P} {s_{C-P}}^{-1} \bmod n, \tag{148}$$

$$({x_{C-P}}', {y_{C-P}}') = u_{C-P1}G + u_{C-P2}Q_C, \tag{149}$$

$${x_{C-P}}' \stackrel{?}{=} r_{C-P} \bmod n. \tag{150}$$

If the verification is passed, the payment information is confirmed by the content administrator, and the smart contracts cpins and cpchk will be sent. The content is as follows (Scheme 13):

```
function insert smart contract cpins(
string cp_id, string cp_detail,
string cp_key, string cp_tsp) {
        count ++;
         cp[count].id = id;
        cp[count].detail = detail;
         cp[count].key = key;
         cp[count].tsp = tsp;
}
sign string c_key (cp_id, cp_detail,
cp_key, cp_tsp);
```

```
verify string c_key (cp_id, cp_detail,
cp_key, cp_tsp);
function check smart contract cpchk(
string cp_id, string cp_detail,
string cp_key, string cp_tsp) {
        return cp_id.exist;
        return cp_detail.exist;
         return cp_key.exist;
         return cp_tsp.exist;
}
```

Scheme 13. The smart contracts cpins and cpchk.

The Proxy calculates:

$$BC_{C-P} = h(r_{C-P}, s_{C-P}), \tag{151}$$

(ID_{BC}, BC_{C-P}) will also be uploaded to the blockchain center.

- Step 7: The Proxy generates a random value k_{P-L} and calculates:

$$z_{P-L} = h(ID_P, M_{P-L}, (C_1, C_2), TS_{P-L}, ID_{BC}), \tag{152}$$

$$(x_{P-L}, y_{P-L}) = k_{P-L}G, \tag{153}$$

$$r_{P-L} = x_{P-L} \bmod n, \tag{154}$$

$$s_{P-L} = {k_{P-L}}^{-1}(z_{P-L} + r_{P-L}d_P) \bmod n, \tag{155}$$

$$Enc_{P-L} = E_{PK_L}(ID_P, M_{P-L}, (C_1, C_2), TS_{P-L}, ID_{BC}), \tag{156}$$

and sends $ID_P, Enc_{P-L}, (r_{P-L}, s_{P-L})$ to the Licensee.

- Step 8: The Licensee first calculates:

$$(ID_P, M_{P-L}, (C_1, C_2), TS_{P-L}, ID_{BC}) = D_{SK_L}(Enc_{P-L}), \tag{157}$$

uses

$$TS_{NOW} - TS_{P-L} \leq \Delta T \tag{158}$$

to confirm whether the timestamp is valid, verifies the correctness of the ECDSA signature, and then calculates:

$$z_{P-L}{}' = h(ID_P, M_{P-L}, (C_1, C_2), TS_{P-L}, ID_{BC}), \tag{159}$$

$$u_{P-L1} = z_{P-L}{}' s_{P-L}{}^{-1} \mathrm{mod} n, \tag{160}$$

$$u_{P-L2} = r_{P-L} s_{P-L}{}^{-1} \mathrm{mod} n, \tag{161}$$

$$(x_{P-L}{}', y_{P-L}{}') = u_{P-L1} G + u_{P-L2} Q_P, \tag{162}$$

$$x_{P-L}{}' \stackrel{?}{=} r_{P-L} \mathrm{mod} n. \tag{163}$$

If the verification is passed, the authorization information is confirmed by Licensee, and the smart contracts plins and plchk will be sent. The content is as follows (Scheme 14):

```
function insert smart contract plins(
string pl_id, string pl_detail,
string pl_tid, string pl_tsp) {
        count ++;
         pl[count].id = id;
        pl[count].detail = detail;
         pl[count].key = key;
         pl[count].tsp = tsp;
}
sign string p_key (pl_id, pl_detail,
pl_key, pl_tsp);
```

```
verify string p_key (pl_id, pl_detail,
pl_key, pl_tsp);
function check smart contract plchk(
string pl_id, string pl_detail,
string pl_key, string pl_tsp) {
        return pl_id.exist;
       return pl_detail.exist;
        return pl_key.exist;
        return pl_tsp.exist;
}
```

Scheme 14. The smart contracts plins and plchk.

The Licensee calculates:

$$BC_{P-L} = h(r_{P-L}, s_{P-L}), \tag{164}$$

(ID_{BC}, BC_{P-L}) will also be uploaded to the blockchain center. Finally, the APP calculates:

$$P_m = C_2 - C_1{}^{-b} \tag{165}$$

to obtain the identity of the digital content successfully. This step is performed automatically by the smart contract, and the Licensee cannot skip the verification process privately.

4. Analysis

In this section, we analyze the requirements of digital rights management as follows.

4.1. Verifiable

Using digital certificate verification can publicly verify the identity of the Licensee, and the authorization information was published based on the openness and transparency of the information on the chain, truly realizing the high efficiency and specialization in the field of digital copyright.

Let's take the message transmitted by the Licensee (L) and Content Administrator (CA) as an example. When CA sends a message signed by ECDSA to L, L first verifies the correctness of the time stamp and signature, then generates blockchain data $BC_{C-L} = h(r_{C-L}, s_{C-L})$, and uses ID_{BC} as an index to upload the blockchain data to the Blockchain Center (BCC). That is to say, after verifying the correctness of the time stamp and signature for each role that receives the message, it also verifies the correctness of the blockchain data generated by the previous role. Therefore, our proposed solution achieves the characteristics of public verification through blockchain technology and ECDSA digital signature.

4.2. Trustless

The identity of the authorized object of digital content is verified by the Digital Content Administrator. The authorization period is controlled by the Digital Content Administrator. The Licensee cannot occupy or transfer privately. Any nodes that participate in the system do not need to trust each other. The operation of the system and operating rules are open and transparent, and all information is open. A node cannot deceive other nodes. In this way, the trust relationship between nodes is realized, making it possible to obtain trust between nodes at a low cost. For example, when Licensee (L) requests digital content authorization from the Content Administrator (CA), CA will send an authorization message to L. This message $P_m = f(ID_{DC}, Vtime)$ contains the digital content ID and the authorization period, and L will be unable to privately occupy or transfer digital content privately.

4.3. Unforgery

Use time stamp and signature mechanism to irreversibly generate a string composed of random numbers and letters for the data placed in each block. This original text cannot be inferred from the string, thus effectively solving the trust problem. After the hash function operation, the messages are described as follows.

$$
\begin{aligned}
z_{L-M} &= h(ID_L, M_{L-M}, Cert_L, TS_{L-M}, ID_{BC}) \\
z_{M-L} &= h(ID_M, M_{M-L}, TID, invoice, TS_{M-L}, ID_{BC}) \\
z_{L-A} &= h(ID_L, M_{L-A}, Cert_L, TS_{L-A}, ID_{BC}) \\
z_{A-M} &= h(ID_A, ID_L, M_{A-M}, Cert_A, Cert_L, TS_{A-M}, ID_{BC}) \\
z_{M-A} &= h(ID_M, M_{M-A}, TID, invoice, TS_{M-A}, ID_{BC}) \\
z_{A-L} &= h(ID_A, M_{A-L}, TID, invoice, TS_{A-L}, ID_{BC}) \\
z_{L-C} &= h(ID_L, M_{L-C}, Cert_L, TID, Cert_{pay}, TS_{L-C}, ID_{BC}) \\
z_{C-L} &= h(ID_C, M_{C-L}, (C_1, C_2), TS_{C-L}, ID_{BC}) \\
z_{L-P} &= h(ID_L, M_{L-P}, Cert_L, TID, Cert_{pay}, TS_{L-P}, ID_{BC}) \\
z_{P-C} &= h(ID_P, ID_L, M_{P-C}, Cert_P, Cert_L, TID, Cert_{pay}, TS_{P-C}, ID_{BC}) \\
z_{C-P} &= h(ID_C, M_{C-P}, (C_1, C_2), TS_{C-P}, ID_{BC}) \\
z_{P-L} &= h(ID_P, M_{P-L}, (C_1, C_2), TS_{P-L}, ID_{BC})
\end{aligned}
$$

The hash value cannot be reversed back to the original content, so this agreement achieves the characteristic that the message cannot be tampered with.

4.4. Traceable

After the digital content is on the chain, the data block containing the copyright information is permanently stored on the blockchain and cannot be tampered with. All transaction traces can be traced throughout the entire process, which can be used as a digital certificate to deal with infringement. For example: When we want to verify and trace

whether the blockchain data between the Licensee (L) and Content Administrator (CA) is legal, we can compare and verify $BC_{L-C} \stackrel{?}{=} h(r_{L-C}, s_{L-C})$ and $BC_{C-L} \stackrel{?}{=} h(r_{C-L}, s_{C-L})$. When we want to verify and trace whether the blockchain data between the Licensee (L) and Proxy (P) is legal, we can compare and verify $BC_{L-P} \stackrel{?}{=} h(r_{L-P}, s_{L-P})$ and $BC_{P-L} \stackrel{?}{=} h(r_{P-L}, s_{P-L})$. When we want to verify and trace whether the blockchain data between the Proxy (P) and Content Administrator (CA) is legal, we can compare and verify $BC_{P-C} \stackrel{?}{=} h(r_{P-C}, s_{P-C})$ and $BC_{C-P} \stackrel{?}{=} h(r_{C-P}, s_{C-P})$.

4.5. Non-Repudiation

The content of the message sent by each role is signed by the sender with its ECDSA private key. After receiving the message, the receiver will verify the message with the sender's public key. If the message is successfully verified, the sender will not deny the content of the message transmitted. Table 1 is an undeniable description of each role in this program.

Table 1. Non-repudiation of the proposed scheme.

Phase \ Item	Signature	Sender	Receiver	Signature Verification
Authentication and issuing invoice phase (direct authorization)	(r_{L-M}, s_{L-M})	L	CA	${x_{L-M}}' \stackrel{?}{=} r_{L-M} \bmod n$
	(r_{M-L}, s_{M-L})	CA	L	${x_{M-L}}' \stackrel{?}{=} r_{M-L} \bmod n$
Authentication and issuing invoice phase (proxy authorization)	(r_{L-A}, s_{L-A})	L	P	${x_{L-A}}' \stackrel{?}{=} r_{L-A} \bmod n$
	(r_{A-M}, s_{A-M})	P	CA	${x_{A-M}}' \stackrel{?}{=} r_{A-M} \bmod n$
	(r_{M-A}, s_{M-A})	CA	P	${x_{M-A}}' \stackrel{?}{=} r_{M-A} \bmod n$
	(r_{A-L}, s_{A-L})	P	L	${x_{A-L}}' \stackrel{?}{=} r_{A-L} \bmod n$
Payment verification and browsing phase (direct authorization)	(r_{L-C}, s_{L-C})	L	CA	${x_{L-C}}' \stackrel{?}{=} r_{L-C} \bmod n$
	(r_{C-L}, s_{C-L})	CA	L	${x_{C-L}}' \stackrel{?}{=} r_{C-L} \bmod n$
Payment verification and browsing phase (proxy authorization)	(r_{L-P}, s_{L-P})	L	P	${x_{L-P}}' \stackrel{?}{=} r_{L-P} \bmod n$
	(r_{P-C}, s_{P-C})	P	CA	${x_{P-C}}' \stackrel{?}{=} r_{P-C} \bmod n$
	(r_{C-P}, s_{C-P})	CA	P	${x_{C-P}}' \stackrel{?}{=} r_{C-P} \bmod n$
	(r_{P-L}, s_{P-L})	P	L	${x_{P-L}}' \stackrel{?}{=} r_{P-L} \bmod n$

4.6. Data Format Standardization

Effectively categorizing digital content and formatting it on the chain helps to effectively manage digital property rights and control the unique authorization power of digital content, and intellectual property rights can be protected. The CA classifies the original multimedia files and encodes them for storage, which will provide fast and consistent authorized content transmission services.

4.7. Timeliness

In our proposed scheme, the Content Administrator (CA) is responsible for the production and management of the digital content property rights and the identity verification of the Licensee (L); the Content Administrator (CA) is also responsible for the issuance of a time-sensitive playback license, and the Licensee's playback key identification code cannot permanently occupy the playback of digital content. The Licensee must obtain the decryption key through the authorization key. However, the authorization key contains the digital content ID and the authorization period. If the authorization period expires,

the Licensee will be unable to obtain the decryption key; that is, it cannot perform digital content playback. Thus, we do not worry about the leakage of digital property rights.

4.8. Decentralization/Distribution

In the proposed scheme, the information handled by each role is signed by the role with a private key, and the circulation of all information is open and transparent. A node cannot deceive other nodes. In this way, the trust relationship between nodes is realized, making it possible to obtain trust between nodes at a low cost. Thus, the proposed scheme achieves decentralization and distribution.

4.9. Sustainability

The proposed scheme provides two kinds of authority mechanisms. It not only helps to translate the field visit museum into an online visit to a museum's digital collections, but also promotes social education and contributes to the sustainable operation of the museum via our proposed method.

5. Discussions and Comparisons

5.1. Computation Cost

Table 2 is the computation cost analysis of this scheme.

Table 2. Computation cost analysis of this scheme.

Phase \ Item	BCC	CA	P	L
System role registration phase	$1T_{Mul}$	N/A	N/A	N/A
Authentication and issuing invoice phase (direct authorization)	N/A	$7T_{Mul} + 3T_H + 2T_{Cmp} + 2T_{Sig}$	N/A	$7T_{Mul} + 3T_H + 1T_{Cmp} + 2T_{Sig}$
Authentication and issuing invoice phase (entrusted authorization)	N/A	$7T_{Mul} + 3T_H + 3T_{Cmp} + 2T_{Sig}$	$7T_{Mul} + 3T_H + 2T_{Cmp} + 2T_{Sig}$	$7T_{Mul} + 3T_H + 1T_{Cmp} + 2T_{Sig}$
Payment verification and browsing phase (direct authorization)	N/A	$9T_{Mul} + 3T_H + 3T_{Cmp} + 2T_{Sig}$	N/A	$7T_{Mul} + 3T_H + 1T_{Cmp} + 2T_{Sig}$
Payment verification and browsing phase (entrusted authorization)	N/A	$10T_{Mul} + 3T_H + 4T_{Cmp} + 2T_{Sig}$	$7T_{Mul} + 3T_H + 3T_{Cmp} + 2T_{Sig}$	$7T_{Mul} + 3T_H + 1T_{Cmp} + 2T_{Sig}$

T_{Mul}: Multiplication operation; T_H: Hash function operation; T_{Cmp}: Comparison operation; T_{Sig}: Signature operation.

Table 2 is the computation cost analysis of all stages and roles in this scheme. We analyze the payment verification and browsing phase (entrusted authorization) with the highest computational cost. The CA requires 10 multiplication operations, 3 hash function operations, 4 comparison operations, and 2 signature operations. The Proxy requires 7 multiplication operations, 3 hash function operations, 3 comparison operations, and 2 signature operations. The Licensee requires 7 multiplication operations, 3 hash function operations, 1 comparison operation, and 2 signature operations. The method we proposed has a good computational cost.

5.2. Communication Cost

Table 3 analyzes the communication cost of this scheme.

The communication cost analysis of each phase in this scheme is shown in Table 3. We assume that the ECDSA key and signature are 160 bits, the asymmetric message or certificate is 1024 bits, and the rest of the message length such as ID is 80 bits. We analyze the authentication and issuing invoice phase (entrusted authorization) with the highest communication cost. The message sent by the system role to the blockchain center includes 1 other message. The message includes 4 ECDSA keys and signatures, 4 asymmetric messages or certificates, and 4 other messages. The total communication cost in the

system role registration phase is 5056 bits, which takes 0.361 ms under 3.5 G (14 Mbps) communication environment, 0.051 ms under 4 G (100 Mbps) communication environment, and takes 0.253 ms under 5 G (20 Mbps) communication environment [34]. The proposed scheme has excellent performance.

Table 3. Communication cost analysis of our scheme.

Item / Phase	Message Length	Rounds	3.5G (14 Mbps)	4G (100 Mbps)	5G (20 Gbps)
System role registration phase	3552 bits	2	0.254 ms	0.036 ms	0.178 us
Authentication and issuing invoice phase (direct authorization)	2528 bits	2	0.181 ms	0.025 ms	0.126 us
Authentication and issuing invoice phase (proxy authorization)	5056 bits	4	0.361 ms	0.051 ms	0.253 us
Payment verification and browsing phase (direct authorization)	2528 bits	2	0.181 ms	0.025 ms	0.126 us
Payment verification and browsing phase (proxy authorization)	5056 bits	4	0.361 ms	0.051 ms	0.253 us

5.3. Comparison

In this section, we compare the related works which involved the blockchain and smart contract technologies in Table 4.

Table 4. Comparison of the proposed and existing digital right management surveys.

Authors	Year	Objective	1	2	3	4	5	6	7	8
Zhao et al. [17]	2019	Proposed a YODA-based digital watermark management system.	N	Y	Y	Y	N	Y	N	Y
Ma et al. [18]	2018	Proposed efficient and secure authentication, privacy protection, and multi-signature-based conditional traceability approaches.	Y	Y	Y	Y	Y	N	N	N
Vishwa & Hussain [19]	2018	Presented a decentralized data management framework that ensures user data privacy and control.	Y	N	N	Y	Y	N	N	N
Ma et al. [21]	2018	Proposed a blockchain-based DRM platform with high-level credit and security for the Content provider (CP), the Service provider (SP), and customers.	Y	N	Y	N	N	Y	N	N
Lu et al. [22]	2019	Proposed a scheme for digital rights management of design works using blockchain.	Y	Y	N	Y	Y	Y	N	N
Ours	2020	Proposed an authorization of the museum's collections.	Y	Y	Y	Y	Y	Y	Y	Y

1: Blockchain-focused, 2: Comparative analysis with other approaches using tables, 3: Authentication, 4: Verifiable, 5: Unforgeable, 6: Traceable, 7: Data format standardization, 8: Combine cash flow, Y: Yes, and N: No.

6. Conclusions and Future Works

Under the guidance of the "activation and reproduction" public resource thinking based on this research, the use of a "digital authorization" model for museums to provide the information needed by online users and increase financial resources will be a new trend for the sustainable development of museum operations in the future. This research aims at explicating how to use the authorization model that is in line with the actual development of the museum itself and proposes an authorization mechanism based on the blockchain technology related to a museum's digital rights, to realize the economic benefits of the museum collection based on cultural dissemination and education of the public, thereby ensuring the museum's income maximization direction for the perfect development of the current museum-digital authorization model.

This research provides museum exploration based on direct authorization and proxy authorization combined with a cash flow payment verification mechanism. The signature and time stamp mechanism of cryptography is applied to achieve a non-repudiation mechanism (Table 1), which combines blockchain and smart contracts to achieve verifiability, non-tampering, and traceability; digital signatures and digital certificates are used to solve the non-repudiation of the cash flow. Table 2 shows that this method has a good computational cost, while Table 3 shows that the solution we proposed has a low communication cost and can improve the effectiveness of authorization. Table 4 shows the comparison between this digital right management and the existing digital right management survey and proposes a complete presentation of the digital rights of the museum in combination with the financial flow. In addition to the realization of museum social education, the increased benefits of digital rights are conducive to the long-term operation of the museum; the sustainable development of the museum is expected.

In the future, the research will focus on the establishment of a promotion platform for the authorization mechanism of the alliance chain museum of blockchain technology, to achieve a win-win situation of resource sharing and economic benefits. Besides, the world organization has made the world economy globalized and the international market integrated. It is foreseeable that international economic and trade disputes will emerge endlessly. Governments of various countries have added or strengthened arbitration regulations to resolve disputes involving various profits as future digital property management. If there is a dispute, it can be resolved through the mechanism of international legal arbitration. This research provides a good foundation for future research on the authorization of the museum collection alliance chain and the dispute resolution arbitration mechanism.

Author Contributions: The authors' contributions are summarized below. Y.-C.W. and C.-L.C. made substantial contributions to the conception and design. C.-L.C. and Y.-Y.D. were involved in drafting the manuscript. C.-L.C. and Y.-Y.D. acquired data and analysis and conducted the interpretation of the data. The critically important intellectual contents of this manuscript were revised by Y.-C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract number MOST 109-2221-E-324-021.

Informed Consent Statement: This study only base on the 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.

Abbreviations

Abbreviations are used in this paper and listed as follows:

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 identity x
k_x	A random value on elliptic curve
(r_x, s_x)	Elliptic curve signature value of x
$M_{x\text{-}y}$	A message from x to y
ID_{BC}	An index value of blockchain message
BC_x	Blockchain message of x
PK_X/SK_X	An asymmetric public/private key
$E_{PKX}(M)$	Use X's public key PKx to encrypt the message M

$D_{SKX}(M)$	Use *X*'s private key *SKx* to decrypt the message *M*
TID	The transaction identity
ID_{DC}	An identity of digital content
key_m	Asymmetric key containing KeyID and Seed
$Cert_x$	A digital certificate of *x* conforms to the X.509 standard
h(.)	Hash function
$A \stackrel{?}{=} B$	Verify whether *A* is equal to *B*

References

1. Parry, R. *Recoding the Museum: Digital Heritage and the Technologies of Change*; Routledge: London, UK, 2007; pp. 58–81.
2. Fenton, R. *Photographer of the 1850s*; South Bank: London, UK, 1988.
3. The Getty Foundation. Available online: https://www.getty.edu/foundation/initiatives/current/osci/ (accessed on 30 November 2020).
4. Creative Economy Report 2010. United Nations Conference on Trade and Development. Available online: https://unctad.org/system/files/official-document/ditctab20103_en.pdf (accessed on 23 January 2021).
5. Chiou, S.-C.; Wang, Y.-C. The example application of genetic algorithm for the framework of cultural and creative brand design in Tamsui Historical Museum. *Soft Comput.* **2018**, *22*, 2527–2545. [CrossRef]
6. UNESCO. Convention for the Safeguarding of the Intangible Cultural Heritage. 2003. Available online: http://unesdoc.unesco.org/images/0013/001325/132540e.pdf (accessed on 30 November 2020).
7. Chang, C.-W.; Wang, S.-I.; Yang, C.-J.; Shao, K.-T. Fish fauna in subtidal waters adjacent to the National Museum of Marine Biology and Aquarium. *P1atax* **2011**, *8*, 41–51. [CrossRef]
8. Liu, M.-C. Image management procedures of the National Museum of Marine Biology and Aquarium. *Museol. Q.* **2013**, *27*. [CrossRef]
9. ARTouch Editorial Department. The Epidemic Is Not Far Away: 1/3 of the US Museums May Be Permanently Closed, and Japanese Exhibitions with No Works. Available online: https://artouch.com/news/content-12951.html (accessed on 26 November 2020).
10. Chen, H.Y.; Wang, H.A.; Lin, C.L. Using watermarks and offline DRM to protect digital images in DIAS. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 529–531.
11. Thomas, T.; Emmanuel, S.; Subramanyam, A.V.; Kankanhalli, M.S. Joint watermarking scheme for multiparty multilevel DRM architecture. *IEEE Trans. Inf. Forensics Secur.* **2009**, *4*, 758–767. [CrossRef]
12. Tsai, M.J.; Luo, Y.F. Service-oriented grid computing system for digital rights management (GC-DRM). *Expert Syst. Appl.* **2009**, *36*, 10708–10726. [CrossRef]
13. Chen, C.L. A secure and traceable E-DRM system based on mobile device. *Expert Syst. Appl.* **2008**, *35*, 878–886. [CrossRef]
14. Chen, C.L. An all-in-one mobile DRM system design. *Int. J. Innov. Comput. Inf. Control* **2010**, *6*, 897–911.
15. Chen, C.L.; Tsaur, W.J.; Chen, Y.Y.; Chang, Y.C. A secure mobile DRM system based on cloud architecture. *Comput. Sci. Inf. Syst.* **2014**, *11*, 925–941. [CrossRef]
16. Hassan, H.E.R.; Tahoun, M.; ElTaweel, G.S. A robust computational DRM framework for protecting multimedia contents using AES and ECC. *Alex. Eng. J.* **2020**, *59*, 1275–1286. [CrossRef]
17. Zhao, B.; Fang, L.; Zhang, H.; Ge, C.; Meng, W.; Liu, L.; Su, C. Y-DWMS: A digital watermark management system based on smart contracts. *Sensors* **2019**, *19*, 3091. [CrossRef]
18. Ma, Z.; Jiang, M.; Gao, H.; Wang, Z. Blockchain for digital rights management. *Future Gener. Comput. Syst.* **2018**, *89*, 746–764. [CrossRef]
19. Vishwa, A.; Hussain, F.K. A blockchain based approach for multimedia privacy protection and provenance. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bengaluru, India, 18–21 November 2018; pp. 1941–1945.
20. Ma, Z.; Huang, W.; Bi, W.; Gao, H.; Wang, Z. A master-slave blockchain paradigm and application in digital rights management. *China Commun.* **2018**, *15*, 174–188. [CrossRef]
21. Ma, Z.; Huang, W.; Gao, H. Secure DRM scheme based on Blockchain with high credibility. *Chin. J. Electron.* **2018**, *27*, 1025–1036. [CrossRef]
22. Lu, Z.; Shi, Y.; Tao, R.; Zhang, Z. Blockchain for digital rights management of design works. In Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 18–20 October 2019; pp. 596–603.
23. American Association of Museums. *Museums for a New Century, a Report of the Commission on Museums for a New Century*; American Association of Museums: Washington, DC, USA, 1984.
24. Ma, Z. Digital rights management: Model, technology and application. *China Commun.* **2017**, *14*, 156–167.
25. Du Toit, J. Protecting private data using digital rights management. *J. Inf. Warf.* **2018**, *17*, 64–77.
26. Mrabet, H.; Belguith, S.; Alhomoud, A.; Jemai, A. A survey of IoT security based on a layered architecture of sensing and data analysis. *Sensors* **2020**, *20*, 3625. [CrossRef]
27. Szabo, N. Smart contracts: Building blocks for digital markets. *EXTROPY J. Transhumanist Thought* **1996**, *18*, 16.
28. Szabo, N. The Idea of Smart Contracts. 1997. Available online: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html (accessed on 26 November 2020).

29. Han, W.; Zhu, Z. An ID-based mutual authentication with key agreement protocol for multiserver environment on elliptic curve cryptosystem. *Int. J. Commun. Syst.* **2014**, *27*, 1173–1185. [CrossRef]
30. Boneh, D.; Lynn, B.; Shacham, H. Short signatures from the Weil pairing. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Heidelberg/Berlin, Germany, 2001; pp. 514–532.
31. Chen, C.-L.; Yang, T.-T.; Chiang, M.-L.; Shih, T.-F. A privacy authentication scheme based on cloud for medical environment. *J. Med. Syst.* **2014**, *38*, 143. [CrossRef]
32. Chen, C.-L.; Yang, T.-T.; Shih, T.-F. A secure medical data exchange protocol based on cloud environment. *J. Med. Syst.* **2014**, *38*, 112. [CrossRef]
33. Blaze, M.; Bleumer, G.; Strauss, M. Divertible protocols and atomic proxy cryptography. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 127–144.
34. Marcus, M.J. 5G and IMT for 2020 and beyond. *IEEE Wirel. Commun.* **2015**, *22*, 2–3. [CrossRef]