International Journal for Housing Science and Its Applications

Publish September 9, 2025. Volume 47, Issue 1 Pages 349-363

https://doi.org/10.70517/ijhsa47128

Design of Copyright Protection and Network Threat Defense Solutions for the Music Performance Industry Based on Blockchain Technology

Kaixi Yu[1] and Xingnuo Du[2,*]

[1] Hainan Tropical Ocean University, Sanya, Hainan, 572022, China

[2] Department of Performing Arts Management, The Graduate School of Culture Technology, Sangmyung University, Seoul Special City, 110-744, Republic of Korea

Corresponding authors: (e-mail: duxingnuo1992@163.com).

Abstract This paper proposes a solution for trustworthy copyright registration of music digital rights using blockchain technology. This solution enables trustworthy copyright registration of digital music and enhances the security of copyright information storage in the music performance industry. The copyright protection system for the music performance industry is designed and implemented on the Hyperledger Fabric platform, and a music performance copyright image encryption algorithm based on wavelet transform and chaos mapping is designed. After generating hash values from the data, it is uploaded to the sample data chain to form a stable blockchain structure. Backup and query functions are performed via Ethereum, and the Camshift algorithm is invoked to track and locate intruders, triggering active warning and defense functions to achieve network threat defense security design. Experimental results show that the copyright registration time for each piece of music increases by approximately 1.9 seconds, and the average feature fingerprint data for each piece of music stored on IPFS consumes approximately 9MB, meeting performance requirements. The use of wavelet transform-based encryption methods enhances the sensitivity of ciphertext, effectively resisting attacks. The final experiment demonstrates the feasibility of the network threat defense solution.

Index Terms blockchain technology, wavelet transform, chaotic mapping, music performance industry copyright, network threat defense.

I. Introduction

In the context of societal and technological advancements, music performance has evolved into a thriving industry, driving economic development. With the rapid progress of technology and the internet, the digitalization of the music performance industry has become an inevitable trend, giving rise to the digital music industry [1], [2]. However, in the digital music era, digitalization has significantly impacted music, leading to growing challenges such as piracy, copyright infringement, and cyber threats, which have exacerbated the difficulties in protecting the rights of the music industry. These challenges manifest in several key areas: first, the unauthorized dissemination of others' musical works and their distribution across other media platforms or websites; second, the unauthorized commercial use of musical works or failure to pay royalties as agreed; third, unauthorized recording and performance activities during music production and performances; fourth, new forms of infringement stemming from AI deepfake technology, such as the forgery of musicians' voiceprints; fifth, distributed denial-of-service (DDoS) cyberattacks, such as hackers obtaining unreleased audio files from music creators [3]-[7]. These actions severely harm the rights of creators and hinder the healthy development of the music industry. By enacting relevant legal provisions and developing protective technologies (such as digital watermarking and digital blind signature technology), copyright protection and network security can be achieved, thereby safeguarding the healthy development of the music industry. This not only ensures that creators and producers receive reasonable compensation for their work but also encourages music creators to continue producing, fostering the prosperity and innovation of music creation [8]-[11]. Panjaitan et al. [12] pointed out that due to factors such as technological advancements, ease of dissemination, lack of unified international laws, and insufficient public awareness, even when a country has clear music copyright protection laws, enforcement efforts remain hindered, and piracy methods continue to evolve.

To protect music copyright and promote the healthy development of the music industry, the music industry has adopted a series of advanced music copyright protection technologies to address this challenge. Duan et al. [13] developed a perception-based attack method for music copyright detection that integrates multiple features, which is effective on media platforms like

YouTube, and designed an audio signal defense strategy based on this. Lin et al. [14] described application-layer DDoS attack behaviors on websites and designed a request rhythm matrix model based on access trajectory data. By identifying anomalies in the rhythm matrix, they detected application-layer DDoS attacks and achieved defense. Cao [15] introduced a cloud music resource security data storage and protection technology, using an anti-virus propagation SDSD algorithm to adjust network and edge device parameters in the edge computing ecosystem to reduce data loss. SPARGER [16] shared a new method for detecting copyright infringement in popular music called "melody testing," which focuses on identifying and comparing melodies in music to provide evidence for determining copyright infringement. Chen et al. [17] developed an active defense-type copyright protection scheme using blockchain technology to achieve music copyright protection. This scheme uses dual authentication to prevent signatures and non-interactive zero-knowledge proof technology to identify illegal copyright transactions, and automatically processes infringement penalties based on smart contracts.

Blockchain technology is a decentralized distributed database technology that has emerged in recent years and is widely used for protecting music copyright. Blockchain technology establishes a network of multiple nodes to record music copyright information on the blockchain, with each node capable of verifying and recording changes to copyright information, ensuring transparency and immutability of copyright information [18], [19]. Kim, A and Kim, M [20] constructed a music distribution model supported by blockchain and smart contract technology, dividing music assets into blocks for distribution and managing music copyrights through distributed ledger technology, enabling music rights holders to effectively receive royalties. As Sharp [21] pointed out, while music copyright associations and music copyright alliances play a role in protecting copyright in the music industry to some extent, the entire music royalty distribution system cannot address issues arising from inconsistent metadata and database storage. Blockchain, smart contracts, and non-fungible tokens (NFTs) make royalty distribution and copyright protection more transparent and intelligent. Li[22] uses artificial intelligence, blockchain, and encryption technology to convert music files into NFTs and activate smart contracts, thereby achieving music copyright protection. Cai[23] employs deep generative adversarial networks, multi-task learning, and blockchain technology to establish a digital music copyright protection system capable of effectively distinguishing between monophonic melodies and complex melodies, maintaining high accuracy even under multi-user operations. Fang[24] utilized deep belief networks to extract and classify features in music data for identifying and tracking copyright-protected music and related transactions. By combining blockchain technology for decentralized management of such transactions, a music copyright management system was constructed. Wen[25] proposed a trustworthy digital music copyright verification scheme based on blockchain technology, maintaining network security for the music copyright management system under a Byzantine fault-tolerant authorization algorithm.

The study first designed and implemented a consortium blockchain-based digital music copyright protection and transaction system using the Hyperledger Fabric platform. The Shazam algorithm was employed to extract musical feature fingerprints that can prove the originality of music. These fingerprints were stored on the InterPlanetary File System (IPFS). Leveraging the robustness and stealthiness of audio watermarking technology, the system provides trustworthy evidence for creators to protect their rights. Subsequently, two different chaotic mappings were combined with wavelet transforms to achieve secure encryption of the original image from a frequency domain perspective. Next, based on the InterPlanetary File System for storing network data, the data is hashed and uploaded to the sample data chain to form a stable blockchain structure. Backup and query functions are implemented via Ethereum to ensure that the system can invoke the Camshift algorithm to track and locate intruders, and activate proactive warning and defense functions when facing malicious tampering or attacks. Finally, simulation experiments were designed to explore the effectiveness of copyright protection in the music performance industry and image encryption.

II. Copyright protection in the music performance industry based on blockchain technology

II. A. Blockchain Technology Principles

II. A. 1) Blockchain Data Structure

Before the advent of blockchain technology, achieving distributed data synchronization was extremely challenging. In a blockchain-based distributed system, n nodes must record the same data entries, which are defined as a ledger. The blockchain data structure, as shown in Figure 1, consists of individual blocks that store data entries. The creation of each block is not completed by a single node acting alone but is instead achieved through a consensus mechanism in a distributed manner. To maintain the logical order of blocks, those synchronized first are placed at the front end of the data structure, with subsequent blocks following in a chain-like structure to ensure ledger consistency. Each block in the chain-like structure also records the hash of the preceding block, thereby anchoring it to both preceding and succeeding blocks.

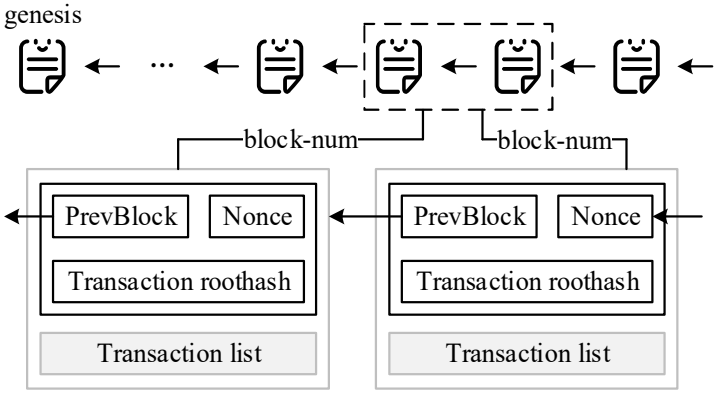


Figure 1: Blockchain architecture diagram

II. A. 2) Blockchain Consensus Algorithms

The Proof of Work (PoW) algorithm is a consensus algorithm based on proof of work. It selects the block-producing node by calculating a random number such that the hash value of the transaction data meets the specified upper limit. The node that first calculates this random number gains the right to produce the new block. Other nodes can easily verify that the block producer has not tampered with the data by checking the hash value provided, and then synchronize the new block to the network. If a blockchain fork occurs, the longer branch of the fork is retained, known as the "longest chain consensus." The POW algorithm has low hash value calculation efficiency, a large number of nodes, and takes a long time to synchronize, making it difficult to apply to notarization scenarios.

POS selects block-producing nodes based on the assets held by nodes, enabling faster generation of block production order. DPOS has fewer block-producing nodes, with on-chain nodes voting using their assets to select 21 block-producing nodes, which synchronize the ledger, resulting in higher efficiency. Within a cycle, the block order is fixed, and each node knows which node will produce blocks at a specific time point. If a node fails to produce blocks during its duty period, it is automatically skipped, and the transaction data during this period is packaged by the next node in line. However, block-producing nodes are highly concentrated, posing a risk of collective tampering. As such, existing blockchain consensus algorithms are not suitable for evidence-of-transaction scenarios.

II. A. 3) Blockchain Application Requirements

To achieve widespread adoption, blockchain-based applications require a sufficiently flexible platform to meet the following requirements:

(1) Support for millions of users. In some cases, the application cannot function unless a critical mass of users is reached

(2) Free to use. Users do not pay to use the platform or benefit from its services. A blockchain platform that is free to use may achieve broader adoption and application.

(3) Security. Building a blockchain-based application platform using Merkle tree structures enables error tracing, as any tampering with a transaction will affect the hash value

(4) Low latency. To handle high user concurrency, timely and reliable responses (with latency not exceeding a few seconds) are essential for a good user experience. Longer latency makes blockchain applications less competitive.

(5) Network performance. The system requires workload distribution across multiple CPUs and computers. As a distributed network, blockchain nodes synchronize data structures every 3 seconds, maximizing performance

II. B. Music Copyright Protection and Trading System Based on a Consortium Blockchain

II. B. 1) System Architecture

The architecture of the music copyright protection and transaction system based on a consortium blockchain is shown in Figure 2. Users register on the client side, and the registered accounts are generated using certificate public keys. Each user obtains a unique digital certificate through the PKI certificate system as an identity marker. The server side includes a contract layer, a network layer, and a storage layer. The contract layer includes identity verification, music work duplication checks, recording music copyright information, and transaction information. The network layer separates the execution of smart contracts from the consensus mechanism to improve efficiency. The transaction process involves endorsement nodes, sorting nodes, and accounting nodes. Endorsement nodes verify transactions, simulate execution, and endorse them; sorting nodes sort transactions and determine their order.

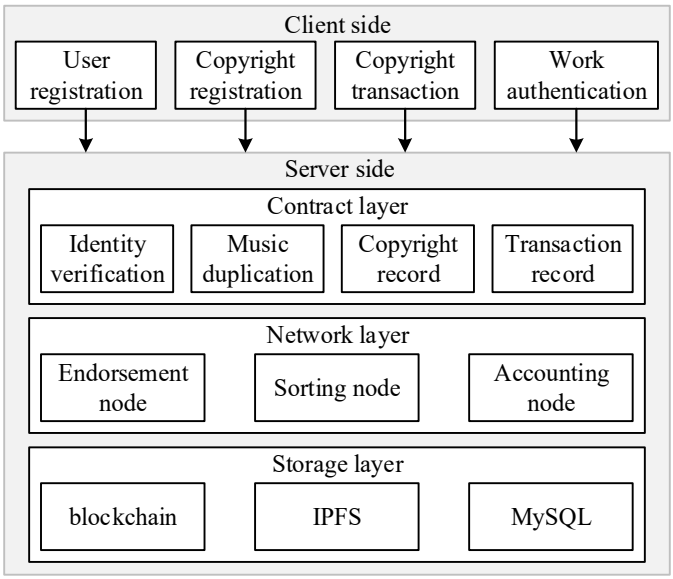


Figure 2: System architecture

II. B. 2) Copyright Registration

In music copyright management, music rights confirmation is the first and most important step. It is essential to ensure the originality of uploaded music to guarantee subsequent copyright transactions and other processes

The system divides users into two categories: administrators and ordinary users. Administrators are responsible for managing the system. Administrators must provide certificates to access the blockchain. Specific operations permitted include:

(1) Adding new smart contracts. Administrators can deploy new smart contracts on the blockchain via API.

(2) Upgrading smart contracts. By changing the version number, administrators can bind the new version of the smart contract to the channel, thereby upgrading the old version to the new version

Ordinary users are the system's end-users, who obtain results by sending requests to the system

After successful registration, users become user nodes, which can upload their original music for copyright registration. The specific process is as follows:

The system uses the Shazam algorithm to extract music fingerprints from uploaded works

$$S(music) \rightarrow fingerprint \tag{1}$$

Convert the original audio data to monaural, and use a first-order digital filter to pre-emphasize the characteristics of the audio signal, enhancing the high-frequency portion to smooth the spectrum and facilitate analysis, as shown in Equation (2), where $x(n)$ is the audio sample value at time n, and a is the pre-emphasis coefficient.

$$y(n) = x(n) - ax(n-1) \tag{2}$$

Since audio signals change over time and their amplitude constantly varies, they must be frame-processed. Through frame processing, the characteristics of the audio signal can be kept essentially unchanged over a period of time. The frame process is actually the addition of a window function, using the window function $w(n)$ multiplied by the signal function $x'(n)$ to form the windowed audio signal.

$$x(n) = x'(n) \times w(n) \tag{3}$$

The window function used in this system is the Hamming window, which is commonly used in frequency domain analysis.

$$w(n) = \begin{cases} 0.54 - 0.46\cos[2\pi n/(N-1)] & 0 \le n \le N-1 \\ 0 & \text{others} \end{cases} \tag{4}$$

Apply a short-time Fourier transform [26] to the audio signal with added windows, as shown in Equation (5), to convert the original time-domain audio signal to the frequency domain and obtain the spectrum diagram.

$$X(m,w) = \sum_{n=-\infty}^{\infty} x(n)w(m-n)e^{-jwn} \tag{5}$$

II. B. 3) Copyright Transactions

This system leverages the immutable and traceable characteristics of blockchain technology to facilitate copyright transactions between users via smart contracts, thereby providing a secure copyright transaction process. As shown in Algorithm A, users can search for music they are interested in, preview it, and make a purchase. Each completed transaction is recorded on the blockchain, and once recorded, the transaction cannot be modified. The transaction record includes the purchaser's ID, the creator's ID, the amount, the IPFS hash address of the music fingerprint, and a timestamp.

$$(hash, user, author, amount, Ts) \rightarrow TX \tag{6}$$

Algorithm a Smart contract for copyright transactions
Input: $transaction$.
Output: Transaction success or failure status.
$\langle hash, user, author, amount, T_s \rangle \rightarrow transaction$
$transactionjson \leftarrow json.Marshal(transaction)$
$err \leftarrow APIstub.PutState(hash, transactionjson)$
If err!=nil
Return Error("stub.PutState err.")
endif
return Success("CreateTransaction success")

Each transaction is transmitted through the P2P network, verified by the consensus mechanism, and finally recorded in the ledger.

II. B. 4) Authentication of Works

Users can upload suspected pirated music files to the system, which will detect whether the files contain platform watermarks. Based on the copyright information and copyright transaction information of the work, the system will determine whether the uploaded music is pirated. Users can also upload music to determine if there is plagiarism or similarity. The system will match the files based on their feature fingerprints and provide similarity results.

After the original creator uploads the music fingerprint to IPFS, the returned hash value, work information, and timestamp are uploaded to the blockchain together. Users can trace the copyright information of the work based on the hash value. The smart contract for copyright queries is shown in Algorithm B.

Smart contract for algorithm b copyright query
Input: $hash$.
Output: Copyright information or failure status.
$copyright, err \leftarrow APIstub.\ GetState(hash)$
If err!=nil
return Error("stub.PutState err.")
end if
return $copyright$

II. C. Music performance copyright image encryption based on chaotic mapping

II. C. 1) Chaotic Mapping

Chaotic systems have many characteristics, such as high sensitivity to initial conditions and fractal properties, which make them widely used in the field of image encryption. This paper mainly uses the following two chaotic mappings:

(1) Nonlinear cubic mapping

The nonlinear cubic mapping is defined as shown in formula (7), where the initial value and parameters are set to $3.3 \le \mu \le 4$, $0 < x_1$, and $n \in N$. The resulting chaotic sequence x_n oscillates irregularly between [−1, 1].

$$x_{n+1} = \mu x_n^3 + (1-\mu)x_n \tag{7}$$

(2) Henon map

The Henon map is a two-dimensional map and is the simplest nonlinear map among high-dimensional chaotic maps. It is defined as shown in formula (8), where the initial values and parameters are set to $1.07 \le a \le 1.4$, $b = 0.3$, and $n \in N$. The resulting chaotic sequences are x_n, y_n

$$\begin{cases} x_{n+1} = 1 + y_n - ax_n^2 \\ y_{n+1} = bx_n \end{cases} \tag{8}$$

II. C. 2) Wavelet Transform Theory

Wavelet transform is a commonly used method in time-frequency domain analysis, superior to traditional Fourier analysis. Its idea is to use a family of functions from a wavelet function system to represent or approximate a signal, and this family of functions is obtained by translating or stretching the basic wavelet function [27]. In the field of image data compression, we usually use wavelet transform to achieve targeted compression of image data, thereby improving the network transmission rate of image data.

Definition of the continuous wavelet transform:

$$W_{\varphi}f(a,b) = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}^{*}(t)dt = \langle f(t),\psi_{a,b}^{*}(t)\rangle \tag{9}$$

In the formula: the window function $\psi_{a,b}^{*}(t) = |a|^{-\frac{1}{2}}\psi(\frac{t-b}{a})$, $\psi(t)$ is the wavelet basis function, the scale parameter $a \in R$, and $a \neq 0$, b is the localization parameter; $\psi_{a,b}^{*}(t)$ is the conjugate of $\psi_{a,b}(t)$.

In computer processing, it is often necessary to discretize continuous wavelets. Specifically, this is done by setting $a \in a_0^j, b = ka_0^j b_0$ ($a \neq 1, k \in R$ and $j \in Z$), to obtain the corresponding discrete wavelet transform:

$$W_{\varphi}f(j,k) = a_0^{-\frac{j}{2}}\int f(t)\psi^{*}(a_0^{-j}t - kb_0)dt = f(t),\psi_{j,k}(t)\rangle \tag{10}$$

In image processing, it is often necessary to perform a two-dimensional discrete wavelet transform on continuous wavelets. Specifically, this is done by setting $b_1 = ka_0^j b_0, b_2 = ma_0^j b_0 (m \in R, j \in Z)$, as shown in the following formula:

$$W_{\varphi}f(j,k,m) = a_0^{-j}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\psi^{*}(a_0^{-j}x - kb_0, a_0^{-j}y - mb_0)dxdy \tag{11}$$

The decomposition process is shown below. Each layer of decomposition produces approximate components and detail components. The detail components include horizontal detail components, vertical detail components, and diagonal detail components.

II. C. 3) Encryption Algorithm Design

The algorithm consists of two processes: scrambling the pixels of the original image and replacing the grayscale values. Assuming that the original image I has a size of m×n (M=m×n), the specific steps of the encryption algorithm are as follows:

(1) Select the p parameter of the two-dimensional chaotic map based on the number of iterations N.

(2) Based on the p parameter and the initial values x1 and y1 of the two-dimensional chaotic map, generate the new initial values X(N+1) and chaotic sequence X of the chaotic map.

(3) Generate the cubic chaotic mapping sequence Z based on the initial value X(N+1). Extract two sub-sequences, sort both sub-sequences, and generate an index number sequence numZ for one of them. Perform a modulo 256 operation on the index number sequence generated for the other sub-sequence to obtain an integer sequence D.

(4) Sort the sequence obtained in step 2, and perform global randomization on each pixel of the original image according to the index number sequence numX.

(5) Perform multi-level wavelet decomposition on the randomized result, and perform randomization again on the final approximation component using the index number sequence numZ obtained in step 3.

(6) Reconstruct the scrambled pixel matrix in the reverse order of the wavelet decomposition to generate a new matrix 10.

(7) Generate a new chaotic sequence d using the Cubic chaotic map based on the initial value $x2$

(8) Use equation (12) to perform amplification, rounding, and modulo operations on the real-valued chaotic sequence d, transforming it into an integer sequence Z

$$Z(i) = ceil(\mathrm{mod}((d(i) | - floor(| d(i) |) \times 10^{16}, 255) \tag{12}$$

(9) Perform the first round of encryption on the pixel values. First, encrypt the first pixel of the plaintext image separately using the following formula:

For the first element: extract an element from the sub-sequence in step 5 (this element is controlled by parameter c, and $0<c<b$), and perform an XOR operation with the last element in the original image.

$$C(1) = I0(M) \oplus D(c) \tag{13}$$

For pixels at any other position in the image sequence, the encryption formula is as follows.

$$C(i+1) = I1(i) \oplus (\mathrm{mod}(Z(i) \oplus C(i), 256)) \tag{14}$$

(10) Perform a second round of encryption on the pixel values.

$$S(1) = D(d) \oplus C(M) \tag{15}$$

$$S(i+1) = C(i) \oplus (\mathrm{mod}(Z(i) \oplus S(i), 256)) \tag{16}$$

(11) Convert the encrypted pixel sequence $\{S(i)\}$ into a two-dimensional matrix to obtain the encrypted image.

III. Design of a blockchain network threat defense solution

III. A. Trusted Rights Management in Music Copyright Models

III. A. 1) Originality review based on audio fingerprinting

In the model, for user-uploaded music works, they must first pass the originality review within the model to proceed with music copyright registration, thereby ensuring the validity of music copyright within the system. For uploaded music works, the first step is to extract the audio fingerprint of the music work, using the Shazam audio fingerprint extraction algorithm.

Assuming that all uploaded music works are recorded at 44,100 Hz, 16-bit precision, and stereo, the data size of 1 second of music in the file is approximately 176 KB. The music file is then divided into data blocks of equal length, each 4 KB in size. After obtaining the frequency domain data of the music file, each data block is converted from the frequency domain data to the frequency domain data using the Fourier transform.

After a user uploads a music work, the system extracts the audio fingerprint of the music work and matches it with the fingerprint database in the system. Only music works that do not have the same audio fingerprint as those in the fingerprint database can be successfully uploaded, thereby effectively ensuring the originality of the music works in the system.

III. A. 2) Generation of Audio Fingerprint Database

In the music copyright protection model, we use audio fingerprint extraction technology to extract audio fingerprint data from music works. We then match the hash key points in the audio fingerprints with those in the fingerprint database to determine the originality of the music works. Theoretically, when there are no matching fingerprint key points in the fingerprint database, it proves the originality of the music works.

The extracted audio fingerprints are used to generate an audio fingerprint database. After each update, the fingerprint database is uploaded to IPFS, which returns a content address. IPFS has the characteristic of one file corresponding to one address, so we only need to ensure that the IPFS address is not tampered with to guarantee the security of the fingerprint database.

As the number of registered musical works increases, the size of the audio fingerprint database grows larger, and storing it on the blockchain becomes too costly. Therefore, we choose to store the latest IPFS address of the fingerprint database on the blockchain, where we can trace the modification history of the fingerprint database.

III. B. Network Threat Defense Workflow

III. B. 1) Data Storage and Transmission Defense

The system collects, transmits, and communicates data in cyberspace through a bottom-up process involving the "IPFS layer—sample data chain layer—Ethereum" workflow. In addition to conventional blockchain-based data transmission and storage, it also has built-in data storage and transmission defense functions. The specific workflow is as follows.

(1) IPFS Layer Defense. IPFS itself has a distributed protocol that can perform diverse encryption on internal network data, such as smart contracts and cryptographic algorithms, effectively ensuring the privacy and integrity of data transmission and storage.

(2) Sample Data Layer Defense. Internal hash pointers are stored in the form of MPT object trees, which offer greater stability and reliability, and information exchange between Ethereum and Polkadot is facilitated through Polkadot's cross-chain technology to prevent network data from being forged or tampered with.

(3) Ethereum layer. Ethereum itself is a mature security ecosystem, with extensive peripheral security maintenance systems built around components such as account trees and smart contracts. Attackers would need to forcefully decompose block hash values to effectively penetrate Ethereum's internal systems and launch attacks, making such attacks extremely difficult.

Of course, the defense mechanism of the aforementioned three-layer network architecture is an internal, passive defense model lacking active monitoring, early warning, and defense capabilities. To enhance system reliability and stability, it is crucial to establish an external intrusion monitoring and active early warning defense system to add an extra layer of security for the system.

III. B. 2) Intrusion Detection Active Early Warning Defense

The Intrusion Detection Active Early Warning and Defense System employs data mining technology to dynamically monitor

network intrusion activities. It utilizes the Camshift algorithm for real-time tracking of intruders. Based on confirmed intrusion types, quantities, and severity levels, the system activates active early warning and defense functions to eliminate intruders and initiate proactive responses. Among these, the Camshift algorithm is the core of this functionality. Compared to conventional network security defense algorithms, the Camshift algorithm possesses stronger adaptive capabilities for early warning and defense. The specific algorithmic approach is as follows: during the process of judging and tracking intruders, the system first selects and judges the intrusion area, which is denoted as (A,B); Next, the target features of the intrusion entities within the region are extracted, and a reverse projection map of the target features is created, denoted as $T(A,B)$; finally, the zero-order and first-order moments of $T(A,B)$ are calculated to determine the horizontal and vertical coordinate information of the center position of the intrusion entity. The formulas for calculating the zero-order and first-order moments of $T(A,B)$ in the Camshift algorithm are as follows:

$$C_{00} = \sum A \sum B T(A,B) \tag{17}$$

$$C_{10} = \sum A \sum B \ AT(A,B) \tag{18}$$

$$C_{01} = \sum A \sum B \ BT(A,B) \tag{19}$$

In the equation: C_{00} represents the zero-order moment of $T(A,B)$, and C_{10} and C_{01} represent the first-order moments of $T(A,B)$.

Based on equations (17) and (18), the expression for the center position information of the invading individuals can be obtained as:

$$(A_c, B_c) = \left(\frac{C_{10}}{C_{00}}, \frac{C_{01}}{C_{00}} \right) \tag{20}$$

In the equation, A_c and B_c represent the horizontal and vertical coordinate positions of the center of the intruding individual, respectively. The essence of using the Cam Shift algorithm for active defense is real-time cyclic scanning, and the entire process requires iterative calculations until the position information of the intruding object converges, at which point the tracking results are output. This "scan-calculate-converge-output" algorithmic approach and monitoring process enables more precise localization and elimination of intruding objects.

IV. System test results and analysis

IV. A. Performance Testing and Analysis

To validate the system's performance, the testing environment used in this paper consists of a Windows 10 operating system configured with an Intel Core i5-7300HQ 2.50 GHz CPU and 16 GB of memory, with a VMware virtual machine running the Ubuntu 16.04 system. The system consists of six major functional modules, with the copyright registration module being the most core and fundamental. This module is the most time-consuming and memory-intensive, as it involves processing music files using the Shazam algorithm and storing music feature fingerprint data. The copyright registration module of the system will be tested next.

This system is built on the VNT Chain test network. Based on performance tests conducted by other scholars, a 10-second audio clip can achieve a high degree of confidence. Therefore, when conducting similarity comparisons, this system will extract the feature fingerprint data of a 10-second clip from the music file to be compared, while the feature fingerprint database constructed by the system will need to extract the feature fingerprint data of the entire music file.

The system performed copyright registration on 50 randomly downloaded music tracks from the internet. Among these 50 tracks, 45 were successfully registered, achieving a success rate of 90%. Analyzing the 5 tracks that failed registration reveals that their durations were all around 5 minutes. Extracting the full feature fingerprint data for these tracks would result in a large data size, triggering the system's over-processing mode and preventing the creation of memory space of that size for the variable. This memory issue can be resolved by modifying system configurations to expand virtual memory or storing feature fingerprint data in batches.

A total of 45 songs were successfully registered for copyright in this system, with the registration time for each song shown in Figure 3. In the figure, the slope of the total time spent on copyright registration is approximately 1.97, meaning that for each song registered, the subsequent copyright registration time increases by approximately 1.97 seconds. The slope of the time spent extracting the feature fingerprint of a 10-second music file is 0.0028, primarily due to system performance fluctuations. Extracting the feature fingerprint of a 10-second music file takes approximately 0.8 seconds. The slope of the time spent by the system to extract the entire feature fingerprint of a music file is approximately -0.0255, which is also caused by fluctuations in system performance. That is, it takes approximately 23 seconds to extract the entire feature fingerprint of each piece of music. The slope of the time spent by the system to perform similarity comparisons is approximately 1.97, which

is almost the same as the slope of the total time spent on copyright registration. That is, the increase in the time required for similarity comparisons is the reason for the increase in the overall copyright registration time.

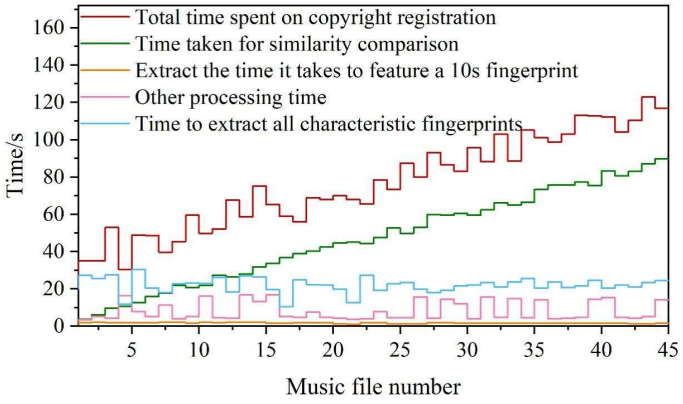


Figure 3: Time spent on copyright registration

The increase in the time spent on similarity comparison has led to an increase in the overall copyright registration time, and the factors contributing to the increase in the time spent on similarity comparison are shown in Figure 4. Observing the figure, it can be seen that the sum of the slopes of the four lines is approximately 1.97, which is the slope of the line representing the time spent on similarity comparison in Figure 3. The time increase caused by obtaining the address of the feature fingerprint on IPFS from the copyright registration contract and the Hash value of the sorted feature fingerprint (one music file corresponds to one feature fingerprint, and one feature fingerprint contains tens of thousands of Hash values) is relatively small. while the system currently uses a multi-process + binary search approach to match the hash values of the feature fingerprints. The time complexity of binary search is logarithmic, meaning that as the data volume increases, the rate of time increase becomes smaller and smaller. Therefore, among these four factors, the most significant factor is retrieving all feature fingerprint data from IPFS. Each successful copyright registration of a music file adds one more feature fingerprint hash address stored on IPFS, and retrieving this hash address takes approximately 1.1 seconds. Future optimizations will focus on data storage and transmission to mitigate the increasing trend in registration time, thereby enhancing the system's practicality.

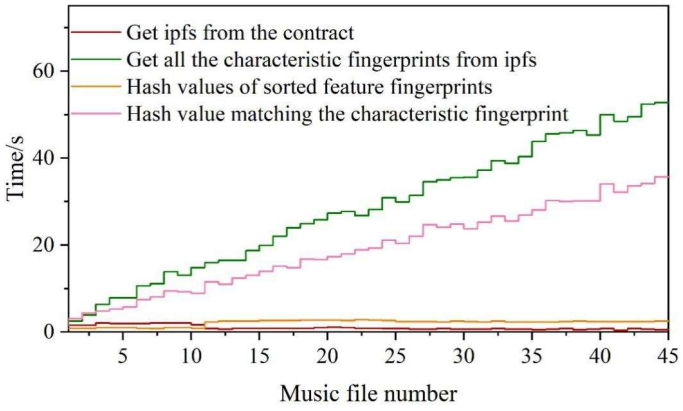


Figure 4: Factors of copyright registration time increase

Among the 45 songs successfully registered for copyright, the storage space consumed on IPFS for the feature fingerprint data extracted from each song is shown in Figure 5, and the number of hash values contained in each feature fingerprint is shown in Figure 6

Among these 45 songs, each song consumes an average of approximately 9 MB of IPFS space. Although 9 MB is 2-3 times the size of the original music file, this is precisely the manifestation of Shazam's algorithm of trading space for time

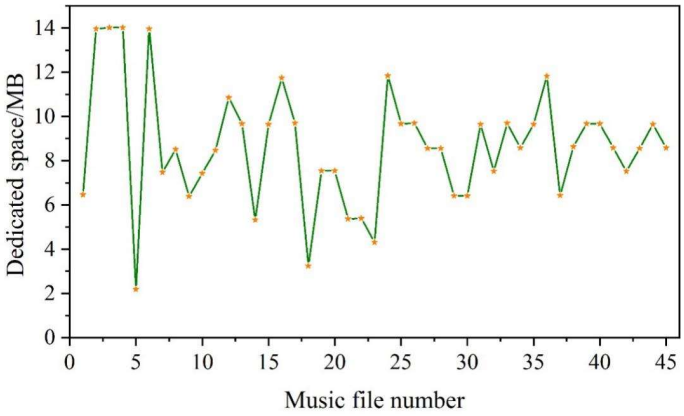


Figure 5: Storage space size costed by each feature fingerprint on IPFS

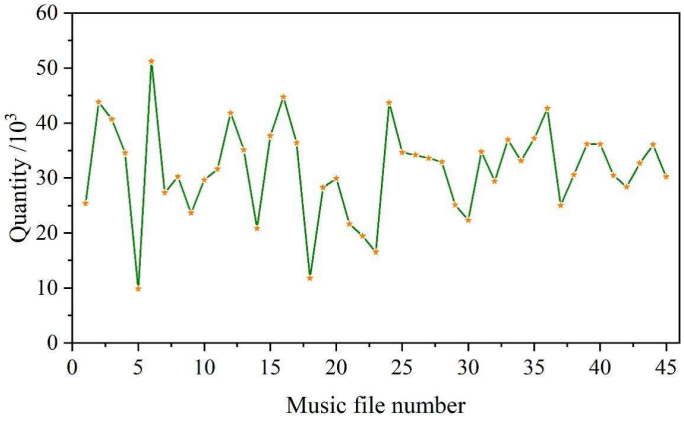


Figure 6: Number of Hash value in each feature fingerprint

After comparing the system described in this paper with those proposed by previous researchers, the results are shown in Table 1. It can be seen that the system described in this paper offers more comprehensive functionality, providing end-to-end copyright services to help users manage the copyright of their musical works more efficiently.

Table 1: System functional module comparison

System	Register of copyright	Copyright trading	Infringement monitoring	Evidence solidifies	Music Ecology
System1	O	O			
System2	O	O			
System3	O				
System4	O				
System5	O				
System6	O				
System7	O	O			
The system in this paper	O	O	O	O	O

IV. B. Copyright Image Encryption Analysis of Music Performances

IV. B. 1) Simulation Results

The simulation test of the encryption system in this paper was conducted on a 64-bit Windows 11 system with a frequency of 2.90 GHz and 8 GB of installed memory, using MATLAB 2016b simulation software to encrypt a 256×256 grayscale image

The initial mapping values were: k=0.8, a=0.6, x=0.5, and y=0.5. The simulation results are shown in Figure 7

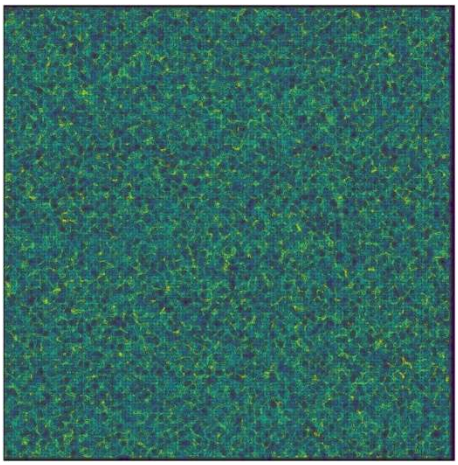


Figure 7: Image encryption effect

IV. B. 2) Correlation analysis of adjacent pixels

Pixel correlation is closely related to image cracking. The higher the correlation, the easier it is to crack. The correlation is calculated using the following formula:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{21}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} [x_i - E(X)]^2 \tag{22}$$

$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N} [x_i - E(x)][y_i - E(y)] \tag{23}$$

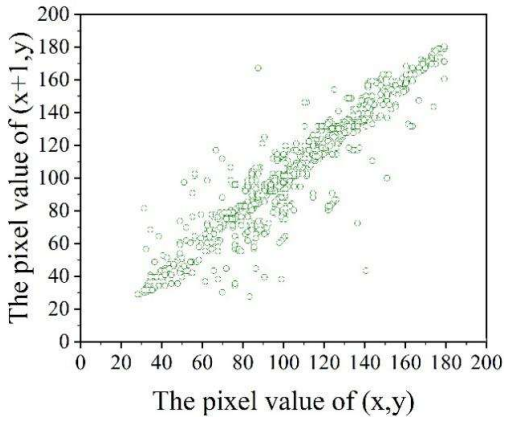
$$r_{x,y} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(x)}} \tag{24}$$

This paper applies the above formulas to plaintext images and encrypted images to obtain the correlation coefficients in the row, column, and diagonal directions, as shown in Table 2. The comparison of encrypted and unencrypted images is shown in Figure 8.

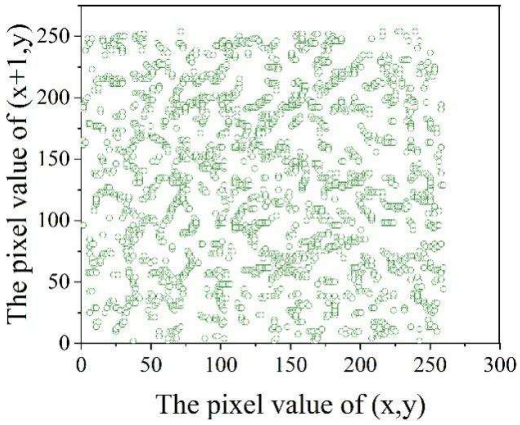
Table 2 compares the correlation coefficients of adjacent pixels between the algorithm proposed in this paper and other methods. As can be seen from the data, the correlation coefficient values of the original image are close to 1, indicating that the original image has strong correlation. By combining the data in Table 2 with Figure 8, it can be intuitively observed that, compared with other algorithms, the encrypted images produced by the algorithm proposed in this paper have smaller differences from 0 in the row, column, and diagonal directions, indicating that the algorithm proposed in this paper can effectively reduce the correlation between plaintext pixels

Table 2: Correlation coefficient comparison between adjacent pixels

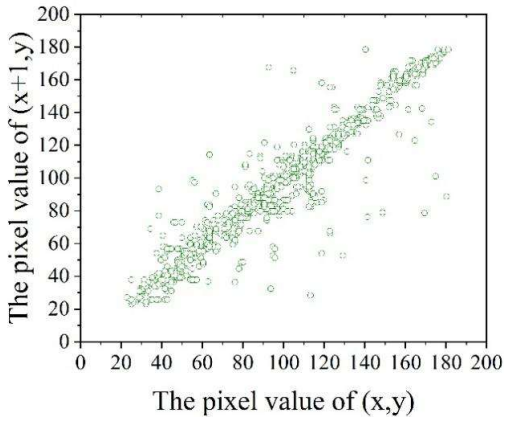
Picture	Direction of travel	Direction columns	In the opposite direction
Original image	0.9152	0.9517	0.9125
Picture 1	-0.0123	-0.0158	0.0193
Picture 2	-0.0284	0.0139	0.0334
This text	-0.0115	0.0042	-0.0035



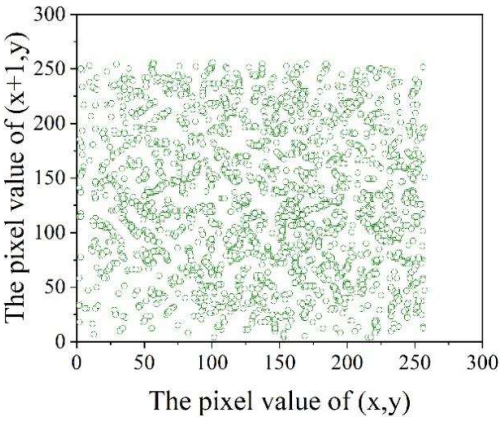
(a) Original line direction



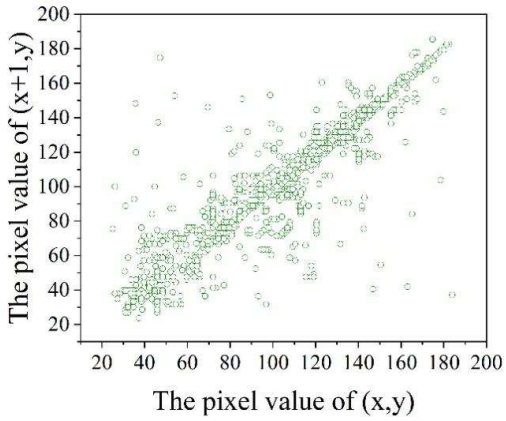
(b) Encryption of the image line direction



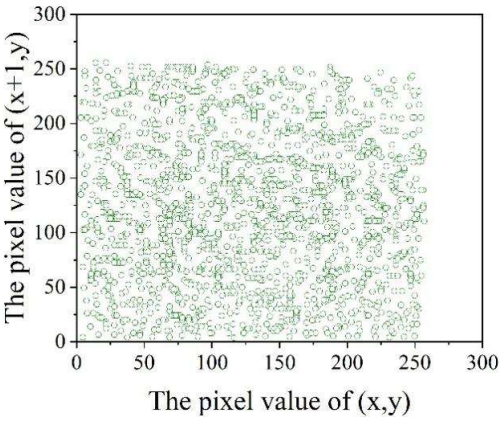
(c) Original image orientation



(d) Encryption of the image column direction



(e) Original image diagonally

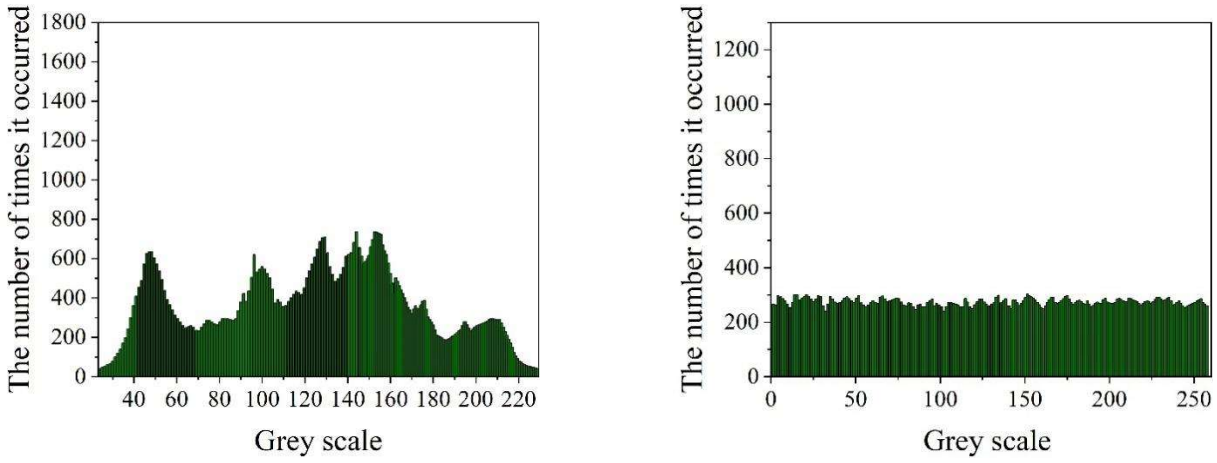


(f) Rotate the encrypted image diagonally

Figure 8: Comparison of correlation between plaintext and ciphertext

IV. B. 3) Histogram Analysis

A histogram statistically analyzes the distribution of gray levels in an image. Figure 9 shows the histogram distribution before and after encryption. The more uniform the histogram distribution, the stronger the image's resistance to attacks. The relatively uniform distribution in Figure (b) indicates that the encryption method described in this paper has strong resistance to attacks.



(a) Plaintext histogram (b) Ciphertext histogram

Figure 9: Histogram distribution before and after encryption

IV. C. Security Defense Results and Analysis

IV. C. 1) Fault tolerance analysis

Fault tolerance is an important goal in distributed systems that reflects system reliability and availability. Specifically, it means that even when a distributed system experiences a failure or contains a certain number of Byzantine fault nodes, the system will not fail and will continue to operate normally. Based on the Camshift algorithm discussed in this chapter, we tested the system's fault tolerance by sequentially setting the number of failed consensus nodes to $f_1 = 0, 1, 2, 3, 4$. The test results are shown in Table 3. Based on the experimental results, we can conclude that when the number of failed nodes in the consensus node set is less than 3, the number of normal nodes is sufficient to ensure that consensus requests in the blockchain network are executed normally. If the master node fails, the Camshift algorithm performs a view change through the view change process and selects a new master node in the new view. However, when the number of failed nodes in the consensus node set reaches 3, the blockchain network operates abnormally, and each request cannot be executed normally, resulting in a system throughput of 0. Through the experiment, it can be determined that the Camshift algorithm can tolerate a maximum of no more than (n−1)/3 failed nodes, and its fault tolerance capability meets the requirements, making it suitable for application in a digital music copyright management system.

Table 3: Camshift algorithm fault tolerance test

Number of invalid nodes	Throughput capacity
0	1726.80995
1	1865.95023
2	1781.1086
3	0
4	0

IV. C. 2) Throughput Analysis

Throughput is used to evaluate a system's ability to process a certain number of requests per unit of time. To a certain extent, it represents the performance of the system. This chapter uses transactions per second (TPS) to express the throughput of the algorithm. The expression formula is shown below:

$$TPS_{\Delta t} = \frac{Transactions_{\Delta t}}{\Delta t} \tag{25}$$

Where Δt represents the consensus interval, i.e., the time interval from transaction issuance to block confirmation, and $Transactions_{\Delta t}$ represents the number of transactions contained in the block during the Δt time interval.

Throughput testing was conducted on the Camshift algorithm and traditional algorithms, without considering network bandwidth issues. The script simulated 1,000 users simultaneously initiating requests, with each user initiating two requests per second. The experiment was conducted continuously for twenty times, with each experiment lasting ten seconds. The

average throughput of the algorithms is shown in Figure 10.

From the experimental results, it can be seen that the average throughput of the Camshift algorithm is 1,237, which is greater than that of the traditional algorithm. Additionally, during multiple experiments, the throughput of the Camshift algorithm exhibited smaller fluctuations and performed more stably, aligning with the practical requirements for use in a music copyright management system

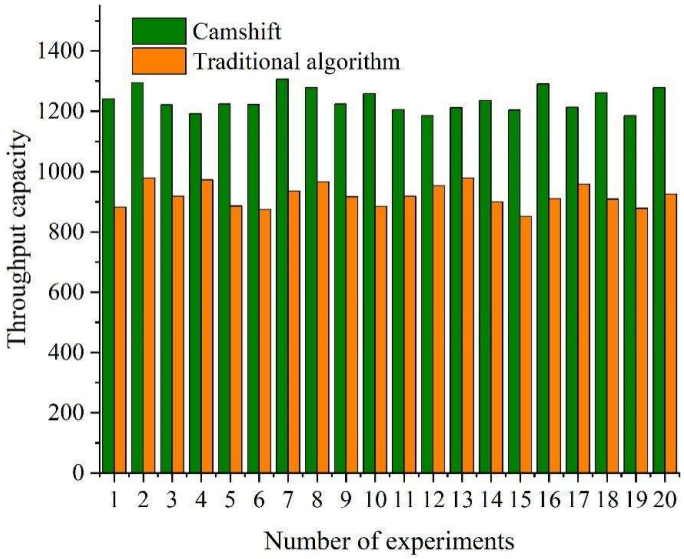


Figure 10: Algorithm throughput comparison

This chapter compares several currently mature blockchain platforms to validate the effectiveness of the Camshift algorithm, with the comparison results shown in Figure 11. As can be seen from the comparison chart, the throughput of the Camshift algorithm has significantly improved compared to several existing mature blockchain platforms, indicating that the Camshift algorithm proposed in this paper can meet the application environment of music copyright management systems.

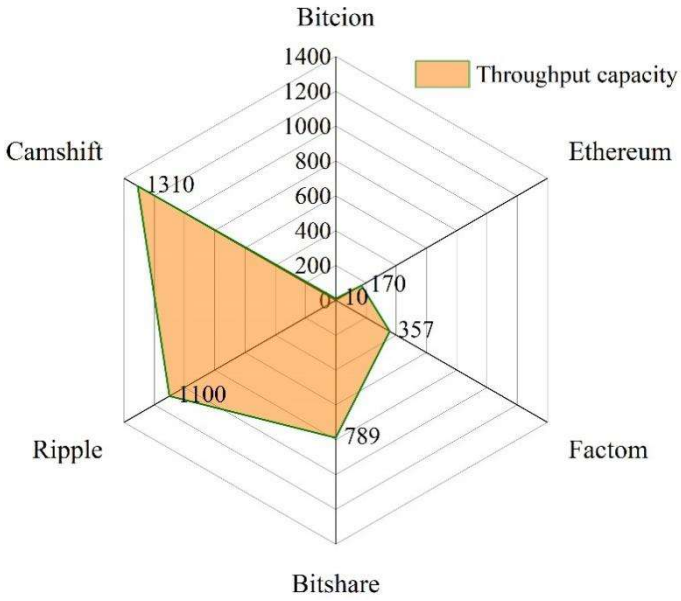


Figure 11: Comparison of throughput between Camshif algorithm and other blockchain platforms

V. Conclusion

This paper addresses copyright issues in the music performance industry by establishing a digital music copyright protection and transaction system based on a consortium blockchain. It then proposes an image encryption algorithm based on wavelet transform and chaotic mapping to encrypt copyright images in the music performance industry. Additionally, a blockchain

network threat defense system is designed using the Camshift algorithm, and the system undergoes performance testing. The results indicate that the established digital music copyright protection and transaction system can provide comprehensive copyright services, helping users manage music copyright more efficiently. The image encryption method using wavelet transform and chaotic mapping achieves high sensitivity of the ciphertext, effectively resisting various attacks. Additionally, the Camshift algorithm proposed in this paper meets the application requirements of music copyright management systems.

Funding

Supported by Scientific Research Foundation of Hainan Tropical Ocean University (No. RHDRCSK202408).

References

[1] Lozic, J. (2019). Digitalization creates a new paradigm of the global music industry: The traditional music industry is under pressure of the streaming platforms. Economic and Social Development: Book of Proceedings, 179-190.
[2] Cross, I. (2023). Music in the digital age: Commodity, community, communion. AI & SOCIETY, 38(6), 2387-2400.
[3] Papies, D., & Van Heerde, H. J. (2017). The dynamic interplay between recorded music and live concerts: The role of piracy, unbundling, and artist characteristics. Journal of Marketing, 81(4), 67-87.
[4] Lee, J. M. (2024). Harmonizing Intellectual Property in the Age of AI-Generated Music. Ohio St. LJ, 85, 953.
[5] Josan, H. H. S. (2024). Ai and deepfake voice cloning: Innovation, copyright and artists' rights. Artificial Intelligence.
[6] Bagal, Y. (2019). Contributory Copyright Infringement in Music Industry: Technological Implications. Journal of Intellectual Property Rights, 24.
[7] Wang, Y., Ma, Y., Ding, J., Sun, X., Wu, D., & Hei, X. (2023, March). Hearing Cyber-Attacks: A Novel Model for Bridging Network Security Situation and Music. In Proceedings of the 2023 5th International Symposium on Signal Processing Systems (pp. 63-70).
[8] Santiago, J. M. (2017). The Blurred Lines of Copyright Law: Setting a New Standard for Copyright Infringement in Music. Brook. L. Rev., 83, 289.
[9] Siregar, R. H. F. (2023). Notes of protection: A comparative analysis of music copyright laws and enforcement. Indonesian Comparative Law Review, 5(2), 115-126.
[10] Baisuni, H., Djulaeka, D., & Sajjad, M. A. (2024). Legal Protection for Unauthorized Copying of Songs on Digital Platforms Through Audio Watermarking Method. JUSTISI, 10(3), 547-564.
[11] Rogers, J., & Sparviero, S. (2025). Copyright, the music business, and the evolution of performing rights organisations. International Communication Gazette, 87(4), 283-290.
[12] Panjaitan, H., Betlehn, A., Situmeang, T., Khan, M. Z. K., & Miraz, M. H. (2024). Music copyright protection in the digital era: Legal framework and strategies for enforcement. Jurnal Hukum UNISSULA, 40(2), 235-257.
[13] Duan, R., Qu, Z., Zhao, S., Ding, L., Liu, Y., & Lu, Z. (2024). Perception-Aware Attack Against Music Copyright Detection: Impacts and Defenses. IEEE Transactions on Dependable and Secure Computing.
[14] Lin, H., Cao, S., Wu, J., Cao, Z., & Wang, F. (2019). Identifying application-layer DDoS attacks based on request rhythm matrices. IEEE Access, 7, 164480-164491.
[15] Cao, H. (2022). Cloud music resources-oriented secure data storage and defense using edge computing. International Journal of System Assurance Engineering and Management, 13(Suppl 3), 1242-1250.
[16] SPARGER, L. (2025). DON'T TRUST THE VIBES: A BETTER TEST FOR POP MUSIC COPYRIGHT INFRINGEMENT. Notre Dame Journal of Law, Ethics & Public Policy, 39(1).
[17] Chen, X., Yang, A., Weng, J., Tong, Y., Huang, C., & Li, T. (2023). A blockchain-based copyright protection scheme with proactive defense. IEEE Transactions on Services Computing, 16(4), 2316-2329.
[18] Arenal, A., Armuna, C., Ramos, S., Feijoo, C., & Aguado, J. M. (2024). Digital transformation, blockchain, and the music industry: A review from the perspective of performers' collective management organizations. Telecommunications Policy, 48(8), 102817.
[19] Zhou, Y., & Huang, F. (2024). Navigating knowledge dynamics: Algorithmic music recombination, deep learning, blockchain, economic knowledge, and copyright challenges. Journal of the Knowledge Economy, 1-25.
[20] Kim, A., & Kim, M. (2020, October). A study on blockchain-based music distribution framework: focusing on copyright protection. In 2020 International conference on information and communication technology convergence (ICTC) (pp. 1921-1925). IEEE.
[21] Sharp, A. (2023). Addressing the Music Industry's Biggest Broken Record: Why Blockchain, Smart Contracts, and NFTs Are an Unmatched Solution to the Music Industry's $424 Million Unmatched Royalty Problem. ABA Entertainment and Sports Lawyer (forthcoming, Summer 2023).
[22] Li, N. (2022). Combination of blockchain and AI for music intellectual property protection. Computational intelligence and neuroscience, 2022(1), 4482217.
[23] Cai, Z. (2020). Usage of deep learning and blockchain in compilation and copyright protection of digital music. Ieee Access, 8, 164144-164154.
[24] Fang, Q. (2024). Designing of music copyright protection system based on deep belief network and blockchain. Soft Computing, 28(2), 1669-1684.
[25] Wen, X. (2023). Application of blockchain technology in copyright protection of digital music information. International journal of grid and utility computing, 14(2-3), 136-145.
[26] Daoui Achraf,Karmouni Hicham,Sayyouri Mhamed & Qjidaa Hassan. (2022). New method for bio - signals zero - watermarking using quaternion shmaliy moments and short-time fourier transform. Multimedia Tools and Applications,81(12),17369-17399.
[27] Han Shaocheng,Lv Mengdie & Cheng Zheng. (2022). Dual-color blind image watermarking algorithm using the graph-based transform in the stationary wavelet transform domain. Optik,268.