TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/14 pp849–861
DOI: 10.26599/TST.2022.9010066
Volume 28, Number 5, October 2023

A Copyright-Preserving and Fair Image Trading Scheme Based on Blockchain

Feng Yu*, Jiahui Peng*, Xianxian Li, Chunpei Li, and Bin Qu

Abstract: With the proliferation of the Internet, particularly the rise of social media, digital images have gradually become an important part of life, and trading platforms have emerged for buying and selling images. However, traditional image trading service providers may disclose users' private information for profit. Additionally, many image trading platforms disregard the fairness of a transaction and the issue of copyright protection after an image is sold. This neglect harms the interests of users and affects their enthusiasm for trading. A secure way to safely transact images is needed. We proposed a copyright-preserving and fair image trading scheme based on blockchain, which combines amplifying locality-sensitive hashing with searchable symmetric encryption to achieve safe image retrieval on blockchain and ensure the credibility of the image retrieval process. Additionally, we use digital fingerprint and watermark technologies to realize the copyright protection of images and use smart contracts to achieve fair transaction processes. The experimental results show that our scheme can protect image copyrights and realize a fair trading process while ensuring efficiency.

Key words: blockchain; copyright-preserving trade; fair trade; tadors code

1 Introduction

With the development of social media (e.g., Facebook and Instagram), the increasing need to store, trade, and share images has become a challenging issue. Image Trading Service Providers (ITSP), a platform for trading images, comes into our life[1–4]. Distributed ITSP has gradually become the direction that researchers pursue. In ITSP, consumers can buy the images that they are interested in, and image owners can obtain benefits, which can inspire them to create images. However, many challenges need to be solved in ITSP because of its centralization, such as transaction fairness, large-scale image storage, and efficient and privacy-preserving image retrieval. In addition, image copyright is the key factor that affects the income of an image owner, and image infringement will cause a great loss to the image owner, so digital copyright protection for images is also a problem faced by traditional ITSP.

Large-scale image sharing and trading help improve social networks; however, images are easily copied, making them easily pirated[5]. The piracy of multimedia, particularly images, is common, but the cost is small. Pirates can even directly download and copy the entire piracy process. However, for resource owners, the frequent piracy of original data will greatly affect their enthusiasm for creation. Protecting the rights and interests of resource owners is actually protecting their enthusiasm for market transactions.

• Feng Yu, Xianxian Li, Chunpei Li, and Bin Qu are with the School of Computer Science & Engineering, Guangxi Normal University, Guilin 541004, China, and with the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China, and also with Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, Guangxi Normal University, Guilin 541004, China. E-mail: yufeng@gxnu.edu.cn; lixx@gxnu.edu.cn; licp821@163.com; 171207887@qq.com.
• Jiahui Peng is with the Guangxi Power Grid Materials Co Ltd, Nanning 530022, China. E-mail: pengjh@gxnu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2022-12-06; accepted: 2022-12-20

In traditional ITSP, platforms have transaction privileges, sellers or buyers need to pay a certain fee to perform transactions, and platforms can collect users' information or even sell it, violating users' privacy. In the traditional image trading platform, pricing is done by the seller, and no bargaining option is available, which is unfair to the buyer.

The current approach to image retrieval is by comparing feature vectors, but plaintext feature vectors divulge image information and can even be used to construct images[6]. Encryption technology is an effective way to protect the privacy of images, but encryption means increasing communication efficiency; the image will lose clarity after encryption, and how to conduct image retrieval in the encryption domain is a popular research work at present. The current encrypted image retrieval scheme is to store the image dataset in a cloud server, and the encrypted feature vector is also stored. The retrieval process is completed by submitting the search request to the cloud server[7]. However, whether the results returned by the cloud server are correct cannot be verified, and the retrieval process cannot be tracked.

Blockchain has natural advantages as a transaction medium because of its decentralized and traceable technology. Data are divided into blocks in chronological order, each block stores all the value exchange information within a period of time, and all nodes permanently store the data within this period. Each node competes for leadership through specific calculations, packages and distributes the information in the next period, and adds specific information to the back of the previous block, forming the blockchain structure. A feature of the blockchain structure ensures data integrity because each block stores all value exchange information before its creation. It also ensures data rigor because once a new block is added to the chain, the previous block cannot be modified. Blockchain realizes a comprehensive distributed mode from recording to transmission to storage. Data transmission, data verification, and data storage are decentralized, and all participants jointly build the database. Blockchain uses asymmetric encryption algorithms to improve the reliability of data.

Many schemes are used to study transaction security and privacy protection on blockchain[8–10], but for multimedia data transactions, copyright protection and retrieval security should also be the key content. At present, digital copyright protection mainly focuses on studying digital fingerprints, which are embedded in multimedia data and extracted to prove copyright ownership in case of piracy[11]. However, the mere use of digital fingerprints cannot effectively prove copyright ownership because of the lack of a trusted depository platform. The emergence of blockchain has solved this problem well. We design a trading scheme with copyright protection and fair trading based on blockchain, implement trusted deposits from a digital fingerprint using a smart contract, and apply commutative encryption watermarking to embed digital fingerprints, ensuring the security of the embedding process. In addition, the image trading scheme that we designed fully considers the security and access control of image retrieval. We use Amplifying Locality-Sensitive Hashing (ALSH) with Searchable Symmetric Encryption (SSE) to achieve safe image retrieval on blockchain, and use a secret sharing scheme combined with zero-knowledge proof to achieve access control of IPFS. Our work mainly includes the following contributions:

(1) Combined with ALSH and SSE, an efficient and secure image retrieval process is realized. Compared with traditional Locality-Sensitive Hashing (LSH), our scheme not only improves the retrieval efficiency but also considers security, and uses a smart contract to complete the entire retrieval process and realize the verifiability of the retrieval. We have tested our scheme on the famous image dataset Caltech256, and the results show that our scheme is feasible.

(2) Our proposed scheme combines Tardos codes and commutative encryption watermarking to protect the copyright of an image after a transaction. The participants can confirm the copyright reliably by extracting the digital fingerprint in the piracy data and comparing it with the record of the smart contract. We used SGX to complete the testing process, ensuring security and reliability.

(3) We have implemented an auction mechanism based on Vickrey-Clarke-Groves in smart contracts that maximizes the expected payment of each bidder and ensures fairness among transaction participants, and all bidding records are traceable.

2 Related Work

Satoshi Nakamoto is believed to be the first researcher to propose the concept of blockchain[12]. The blockchain's consensus algorithm makes the blockchain difficult to tamper with unless a miner can control 51% of

the network's computing power, which is considered impossible. Additionally, the blocks contain the encrypted hash of the previous block and other key information between them, which constitutes a data structure that cannot be modified.

A smart contract[13] is an assembly language program on a blockchain and is often considered an automatic guarantee account, for example, where the program releases and transfers funds when certain conditions are met. Some studies[14, 15] used a smart contract combined with SDN to implement a multi-scenario task offloading strategy.

In terms of a verifiable image retrieval scheme, Li et al.[16] proposed to use simhash to build a feature vector index to compress the retrieval time. Simhash compresses a high-dimensional feature vector into 0, 1 encoding, which not only greatly improves the retrieval speed, but also results in plummeting retrieval accuracy due to the compression of the dimension of the feature vector. Similar to other work[17–19], this scheme realizes the verifiable retrieval of images using smart contracts combined with SSE on the blockchain. SSE implements the process of image retrieval under the condition of encryption. A smart contract ensures the verifiability of retrieval and can show good results in large-scale image retrieval with low dimensions.

Current work mainly focuses on Digital Rights Management (DRM) in copyright protection with multimedia data, digital fingerprints, and digital watermarks[20]. DRM achieves copyright protection through the entire process of image management from production to sale, while digital fingerprints aim at the copyright protection of the transfer process. We mainly study the work of digital fingerprints in copyright protection. Digital fingerprints include symmetric fingerprints[21], asymmetric fingerprints[22], and anonymous fingerprints[23]. The three types of fingerprint forms have their advantages and disadvantages. For different transaction scenarios, different fingerprint coding methods can be selected. At present, some studies combine blockchain and digital fingerprints to conduct trusted copyright protection[24–26].

Blockchain tamper-resistant features are rarely studied for copyright protection. Instead, the research is mainly focused on combining chain blocks with cryptography[27], such as zero-knowledge proof, oblivious transfer, and RSA encryption. However, using cryptography for copyright protection will lead to decreased efficiency of communication[28]. At present, some studies use the trusted execution environment SGX to complete the copyright verification process, which reduces the problem of low cryptographic communication efficiency[29].

Our work focuses on retrieval security in image transactions and copyright confirmation after transactions. Although some studies are similar, they disregard the confidentiality and efficiency of smart contracts. In addition, most of these studies use cryptography technology in copyright protection and disregard the popular digital fingerprint and blockchain combination.

3 Preliminary

3.1 Locality sensitive hashing

$H(x)$ represents the hash transformation of sample x, $d(s;k)$ is the distance between two arbitrary samples s and k, d_1 and d_2 are two constant values, with $d_1 \leqslant d_2$, and p_1 and p_2 are two constant values between 0 and 1, $P[H(s) = H(k)]$ is the probability of $H(s) = H(k)$. If $H(x)$ meets the following two conditions, it's called a locality-sensitive hash function[30],

(1) $d(s;k) \leqslant d_1$ and $P[H(s) = H(k)] \geqslant p_1$;
(2) $d(s;k) \geqslant d_2$ and $P[H(s) = H(k)] \leqslant p_2$.

3.2 Searchable symmetric encryption

The single-user SSE scheme comprises users and servers. Suppose that Δ is a keyword dictionary and $D \in 2^{\Delta}$ is a file set. The user wants to store the file set D on the server. Additionally, the server can provide a search service for dictionary[31]. We use an index-based SSE scheme, which is essentially a polynomial calculation, to construct the equation: $SSE = (Gen, Enc, Trpdr, Search, Dec)$. The steps are described in detail below.

$K \leftarrow Gen\left(1^{\lambda}\right)$: A probabilistic algorithm for key generation run by the user to establish the system, with the security parameter λ as input and the output key K.

$(I, c) \leftarrow Enc(K, D)$: A probabilistic encryption algorithm run by the user with key K and a collection of files $D = (D_1, D_2, \ldots, D_\iota)$ as input, it generates a security index I and a series of ciphertext $c = (c_1, c_2, \ldots, c_\vartheta)$.

$t \leftarrow Trpdr(K, w)$: A deterministic algorithm run by the user, it outputs a trap gate t based on the desired keyword w and key K as inputs.

$X \leftarrow Search(I, t)$: A deterministic algorithm run by the server, it finds the file containing the keyword w

in file set D according to index I and trap t, and returns the file identifier set X.

$D_i \leftarrow Dec(K, c_i)$: A deterministic algorithm run by the user, it obtains the corresponding ciphertext according to the identifier in X, and decrypts and outputs the final plaintext file D_i with key K.

3.3 IPFS

IPFS was created to replace the HTTP protocol, which is inadequate for today's rapidly evolving distributed systems. IPFS is a distributed storage protocol using the P2P mode to transmit data. Compared with the traditional distributed storage architecture, IPFS is faster, has higher throughput, and uses a directed acyclic graph to achieve an addressable storage architecture. Moreover, IPFS uses distributed hash tables to realize more diversified storage, and there is no need to worry about system failure caused by an IPFS outage[32].

3.4 Auction rules

In some specific scenarios, the rules of auction can be redesigned and improved to solve some specific problems. For example, to avoid the loss of the seller, a reserve price can be set. When the highest bid in the auction does not exceed this price, the auction will not be closed[33]. In addition, bidders can be required to pay a fee before participating in an auction, which can slightly increase the cost for bidders to improve their enthusiasm for participating in the auction and increase the probability of auction success.

Direct mechanism: The direct mechanism comprises two rules: bidders give the estimated price of the subject matter at the same time, and the sum of the winning probabilities of all bidders must be less than or equal to 1. The direct mechanism does not require bidders to give their bid for the subject but the valuation of the subject. Then, the auction facilitates the random selection process to determine the winner and the price. Notably, the bidder with the highest bid does not necessarily win because of the random selection function.

Manifest principle: Any Bayesian Nash equilibrium of any Bayesian game can be expressed as an incentive-compatible direct mechanism. The so-called incentive compatibility means that in the above direct mechanism, bidders have no incentive to offer a price different from their true valuation; that is, the price they offer is their true valuation. The point of the manifest principle is that the effect of any auction mechanism can be transformed by careful design into an incentive-compatible direct mechanism.

4 Proposed Scheme

4.1 Attack model

We describe the problem definition in detail and list the attack models. The first is the attack model of privacy protection.

A leakage of the image feature vector in plaintext may allow the image information to be reverse deduced.

The image retrieval process may reveal users' preference information.

The contents of ledgers and smart contracts are open to the entire Internet, which can reveal the private information of users.

The second is the attack model of copyright protection.

The anti-collusion attack code assigned by the buyer is sent to the seller as plaintext, and the seller can frame the innocent buyer by constructing the code word.

Some buyers conspire to attack and modify the code word piracy.

The third is the attack model of fair trading.

The buyer does not pay after receiving the image sent by the seller (i.e., the delivery process cannot be completed after making the payment).

Because of the openness of smart contracts, the bargaining process between buyers and sellers leads to the other party knowing the price.

4.2 System model

The scheme that we proposed comprises four entities: Data Owner (DO), authorized Search User (SU), Smart Contracts (SCs), and IPFS. The DO submits the encrypted index and the information to the SC. Hash is the address returned by IPFS. The detailed steps are shown in Fig. 1. The SU wants to buy data, and he/she should submit the set to the SC. The keyword (token) is the value of ALSH of the image he wanted to buy. After obtaining the SU's public key from the SC, the DO needs to preset the price (detailed in the following chapter) and upload the picture to IPFS. IPFS returns the image storage address, and DO encrypts the address and stores it in the ledger. The SU obtains the address of the image and downloads the image to complete the transaction. Our trading scheme mainly includes privacy-preserving image retrieval, a copyright-preserving scheme, and fair trading, which will be introduced in detail as follows.

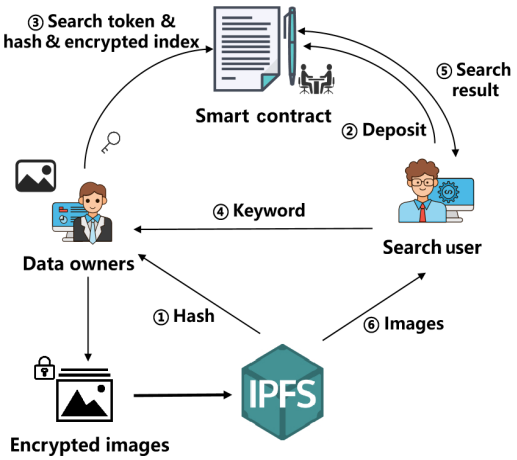

Fig. 1 Scheme framework. There are four main entities, DO, SU, IPFS, and SC. Steps ①–⑥ represent the approximate execution order.

4.2.1 Privacy-preserving image retrieval scheme based on blockchain

This section will introduce our proposed Privacy-Preserving Image Retrieval (PPIR) scheme based on blockchain in detail. PPIR is mainly divided into two schemes: the construction of an index based on ALSH and searchable encryption based on blockchain.

The main process of the scheme is shown in Fig. 2. After the SU submits the deposit, it initiates an image query request with the SC, and then the SC uses the search token to conduct a trap function search on SSE, and finally returns the image search results.

PPIR is mainly divided into three processes: feature vector extraction, index build, and search on the blockchain.

(1) Feature vector extraction

PPIR adopts a compact representation to prove its effectiveness for specific object retrieval[34]. It inputs the image set P_l, $l \in \{1, 2, \ldots, \tau\}$ and outputs 3-dimensional vectors of $W \times J \times M$ dimensions, M is the last layer of the map. The detailed steps are as follows:

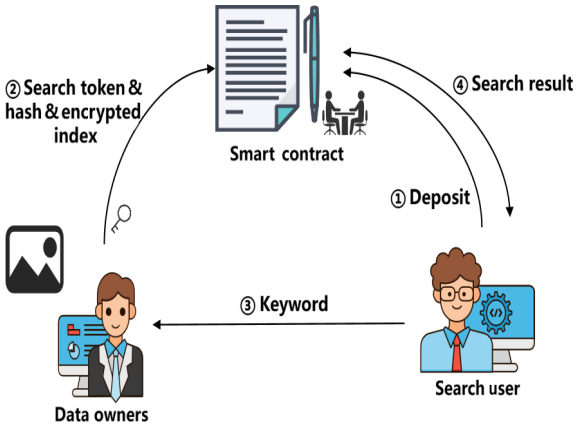

Fig. 2 Framework of the retrieval scheme.

Step 1: PPIR uses max-pooling to construct the Maximum Activations of Convolutions (MACs) as follows:

$$f = [f_1 \cdots f_k \cdots f_K]^{\mathrm{T}}, \quad f_k = \max_{x \in \xi_m}(x \cdot \P) \tag{1}$$

where f is the set of feature vectors. In the networks, f is equal to 2048, which makes it a compact image representation. ξ is the set of all $W \times H$ activations for the feature map. ¶ is the indieator function.

Step 2: PPIR uses a two-branch network, each branch adopting a clone of the other branch, and they share the same parameters. We trained the image dataset by inputting pairs (j, k) and labels $Y(j, k) \in 0, 1$, and set the contrastive loss $L(j, k)$ as

$$L(j,k) = \frac{1}{2}(Y(j,k)\|\bar{f}(j) - \bar{f}(k)\|^2 + (1 - Y(j,k))(\max\{\bar{f}(j) - \bar{f}(k)\|\})^2) \tag{2}$$

where $\bar{f}$ is the l_2-normalized MAC vector of the image, and we trained the model by deep residual network, named resnet 50. The resulting initial feature vector dimension is 2048, which can be compressed by compression methods, such as PCA.

(2) Index build

To increase the efficiency of the search, PPIR uses an improved LSH function to construct the index. The accuracy of the traditional LSH function is insufficient, and the probability of similar images being mapped to the same bucket table is small. We improve the accuracy by reducing the false negative rate; that is, we build more appropriate LSH functions to optimize the bucket table, specifically as follows:

Step 1: The improved LSH is actually an optimization of the bucket table; thus, the same image has a higher probability of entering the same bucket table. To achieve this goal, you need to use the hash function family ψ to handle all the feature vectors $\{f_i\}_{i=1}^{n}$ of the image. Then, h LSH functions are selected from the same LSH function family. It is important to note that the value of h should be determined according to the length of the fingerprint value. A larger h does not mean higher accuracy. We will simulate and explain the value of h in the experiment. The function $H(z) = H(r)$ is true when $H_h(z) = H_h(r)$, $H(z)$ and $H(r)$ represent the hash values of the feature rectors of the two images z and r. It also means that when the LSH hash values obtained

by the feature vectors of two images are identical, the two images (two feature vectors with a high probability of being similar) will be mapped to a bucket table; otherwise, they will not be mapped to a bucket table. The purpose of the above operation is to increase p_1 and decrease p_2, that is, to reduce the false negative rate.

Step 2: Many hash values are obtained in the above steps, and the hash table complexity is high when large-scale data must be processed. To reduce the redundancy of the hash table, PPIR performs XOR to reduce the dimensions of the hash, the value of dimensionality reduction $Hash_{new}$ is finally obtained in the following:

$$Hash_{new} = (H_1)\mathrm{XOR}(H_2)\cdots\mathrm{XOR}(H_h) \quad (3)$$

Notably, this step is the XOR processing after the h-th LSH hash processing of a feature vector. Multiple hash values of an original feature vector are merged into one, greatly reducing the number of hash values of the bucket table.

Step 3: Finally, bucket tables must be constructed. PPIR needs multiple bucket tables to store the same image encoding, denoted as $\{BKT_b\}_{b=1}^{N}$, where N represents the number of buckets. PPIR can be used to put the feature vectors $\{f_i\}_{i=1}^{n}$ of similar images processed by the improved LSH function into the same bucket; thus, the feature vectors in a bucket are similar. The index structure is shown in Table 1.

(3) Search on blockchain

The data owner uploads the LSH index through the SC. Each search of the SU actually uses the improved LSH function for feature vector mapping to obtain the index value and compare it with the ledger. PPIR uses the SC to build SSE search steps as follows:

Setup $(1^{\lambda}) \rightarrow (sk, pk)$: Use bilinear pairings $e: G_1 \times G_1 \rightarrow G_2$ to build two hash functions: $H_1: \{0,1\}^* \rightarrow G_1$ and $H_2: G_2 \rightarrow \{0,1\}^{\log p}$, where p is a generator. After inputting parameter λ, output public key $pk := [g, h = g^{\alpha}]$ and secret key and $sk := \alpha$.

Enc $(pk, w, \{P_l\}_{l=1}^{n}) \rightarrow (c, \gamma)$: To construct the encryption index, the SU inputs the public key of the image to be retrieved, inputs a random number $r \rightarrow Z_p^*$, calculates $t = e(H_1(w), h^r)$, outputs $c = [g^r, H_2(t)]$, and obtains the ciphertext corresponding to the keyword searched and the encryption index γ.

TdGen $(sk, f_v) \rightarrow (t_q)$: Generate the search token t_q, input the feature vector of the image f_v and private key sk, and output $t_q = H_1(f_v)^{\alpha}$.

Search $(t_q, \gamma) \rightarrow (hash, \perp)$: Input two parameters γ and t_q, assuming the equation $H_2(e(t_q, \gamma)) = \perp$ is true, then return the IPFS address hash of the top-U images with the highest similarity and download the corresponding encrypted image $\perp$ with address $hash$.

Update $(t_q, \gamma) \rightarrow (\gamma')$: Update requires inputs t_q and γ, data owner sends t_q to the SC to update index. The feature vector mapping is calculated using the improved LSH algorithm and put into a similar bucket table. The final output is γ'.

Table 1 Index structure of ALSH.

Bucket	Index of ALSH
Bucket 1	$\mathrm{ALSH}_1\ \mathrm{ALSH}_5\ \mathrm{ALSH}_2\ldots$
Bucket 2	$\mathrm{ALSH}_3\ \mathrm{ALSH}_7\cdots$
Bucket 3	$\mathrm{ALSH}_4\ \mathrm{ALSH}_9\cdots$
⋮	⋮
Bucket n	$\mathrm{ALSH}_{50}\ \mathrm{ALSH}_6\ \mathrm{ALSH}_{10}\cdots$

4.2.2 Blockchain-based copyright-preserving scheme of images

The SU submitted the set $\{ID_{FP}, Des_{SU}, encFP_{SU}\}$ to the SC to initiate the purchase request, $\{ID_{FP}\}$ is the serial number of trading, facilitating rapid positioning, Des_{SU} is the SU search data description, and $\{encFP_{SU}\}$ is the encrypted fingerprint value of the embedded image, which is the Tardos code[12]. Algorithm 1 shows the process of fingerprint generation.

The DO must be registered in the blockchain to be traded and must upload $\{FP_{DO}, dec_{DO}, price\}$ through SCs. FP_{DO} is the sequence of fingerprints for each transaction, comprising the Tadors code, des_{DO} is the data description of the sale, and *price* is the price of the image.

The DO inserts the generated fingerprint FP_{DO} into the image set P_l, $l \in 1, 2, \ldots, n$, using Commutative Watermarking and Encryption (CWE)[35]:

$$\bigoplus_{W_k}(E(P), FP) = E(\bigoplus_{W_K}(P, FP)) \quad (4)$$

where $E(\,)$ is the encryption function, P is the key of symmetric encryption, FP is the embedded

Algorithm 1 Fingerprint generation

Input: Number of users: Num_u, safety parameter: ϵ, fingerprint length: $Length$

Output: Tardos codes: $F_{i,j}$, probability: p_i

Begin

Select p_i on the open interval of the probability distribution P;

for all $1 \leqslant i \leqslant Length$ and $1 \leqslant j \leqslant Num_u$;

Calculate $\Pr[F_{i,j} = 1] = p_i$, $\Pr[Fi; j = 0] = 1 - p_i$;

return Tardos codes;

$FP = F_{i,j}$, where $i \in Length$, $j \in Num_u$, and probability p_i;

End

fingerprint (Tardos codes), and $\biguplus_{W_k}$ is the watermark mark. CWE is a combination of cryptographic and digital watermarking technology, used for multimedia information security and copyright protection.

If the data owner finds that the original image is pirated, the fingerprints in the pirated image must be extracted first, and then the extracted fingerprints are submitted to the SC running in SGX for comparison. The main steps are as follows:

Step 1: Use the public key of the SU to encrypt FP_{DC} and obtain $E(FP_{DC})$.

Step 2: If $FP_{DC} = E(FP_{DC})$, then there is piracy. The scheme we proposed is to identify at least one conspirator for collusion identification (detailed steps are in Algorithm 2), calculate the top-k conspirators according to the score, and punish them. S_i is the set after the alignment score is calculated using the Tardos code[36]. Algorithm 2 shows the process of dection.

4.2.3 Fair image trading scheme based on blockchain

Before the price game, the trading parties exchange the shared key through Diffie-Hellman, x and y are random numbers, the process of verifying the secret key is shown as follows:

$$E(a^{xy}, Sig(SK_{DO}, H(a^x)||H(a^y))) \tag{5}$$

$$E(a^{xy}, Sig(SK_{SU}, H(a^y)||H(a^y))) \tag{6}$$

where a is the primary root of the discrete logarithm, $Sig(\,)$ indicates the signature of information, $H(\,)$ is a hash function, SK_{DO} and SK_{SU} are the private keys for DO and SU, respectively. After the two parties exchange the public key through the SC, the public key is obtained by calculating whether the results of Formulas (5) and (6) are the same.

Process of fair trading: The cost of collecting images increases with the image scale, including time, energy, and hardware resources. In this paper, the collection cost function is assumed to be a monotonically increasing convex function, and the data cost function $s(q)$ is defined as

$$s(q) = \varphi \times q \tag{7}$$

where $\varphi \geqslant 0$ is the cost of a image, and q is the number of images.

Suppose there are b SUs, each of which has an independent estimate of the value of the data. For a SU, this valuation of the image is expressed as v_i in this paper. All SUs submit their bids, and after receiving all of these bids, the DO calculates the distribution of the data and the corresponding payment price p_i through a specific mechanism. Let η be the image assigned by SU and p_i be the corresponding price paid. In this way, the utility functions of the DO and SU can be expressed as follows, respectively:

$$y_0 = \sum_{i=1}^{n} x_i p_i - s(q) \tag{8}$$

$$y_i = v_i \eta - p_i \tag{9}$$

In Eq. (8), the DO maximizes its benefit by setting the price and the number of auctions, and in Eq. (9), the SU will also submit its bid to maximize its benefit.

Finally, SU decrypts the downloaded image from IPFS.

In the process of the transaction, we propose a scheme to consider access control. The main control is the hash address returned by IPFS, and the main process is shown in Fig. 3. Even if the transaction is completed, the SU cannot directly obtain the stored hash of the image. The hash is obtained after the DO stores the image to IPFS before the transaction is divided into υ shared values using the SSS algorithm: $split = \{s_1, s_2, \ldots, s_\upsilon\}$. The shared values are individually encrypted and sent to the SC for saving.

Algorithm 2 Algorithm of detection with the Tardos code

Input: Fingerprints extracted from pirated data: x_i, probability: p_i, fingerprint length: *Length*, Fingerprint saved on chain: $F_{i,j}$

Output: Score after smart contract comparison: S_i

Begin

$x_i \leftarrow$ Fingerprint extraction algorithm;

Calculate $S_i = \sum_{i,j=1}^{Length} S_{i,j} = \sum_{i,j=1}^{Length} x_i U_{i,j}$;

//$S_{i,j}$ indicates the i-th bit code word of the j-th participant;

$U_{i,j}$ is the set of all users in the system,

$$\text{where } U_{i,j} = \begin{cases} -\sqrt{\frac{p_i}{1-p_i}}, & \text{if } F_{i,j} = 0; \\ \sqrt{\frac{1-p_i}{p_i}}, & \text{if } F_{i,j} = 1; \end{cases}$$

End

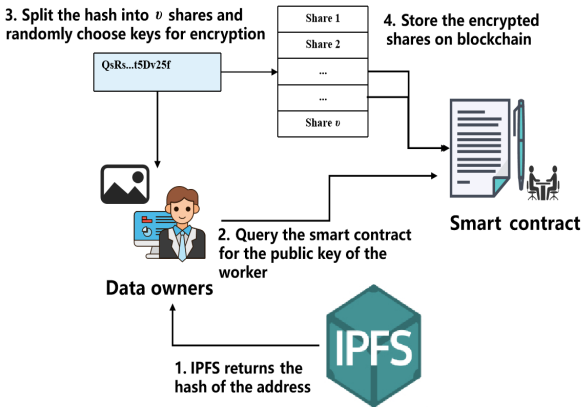

Fig. 3 Framework of the image download.

We used the SSS algorithm based on Pedersen's promise and could verify it. It will return a hash value of h from IPFS as a constant term structure polynomial, $F(x) = s + a_1\beta + \cdots + a_{k-1}\beta^{\epsilon-1}$, and then calculate $TX_i = F(\beta_i), 1 \leqslant i \leqslant n$. After the SC encrypts and stores each TX_i, it calculates $A_i = a_i G, (0 \leqslant i \leqslant k-1)$ and broadcasts it to the entire network.

The worker node validates $TX_i \times G = \sum_{j=0}^{k-1}(x_i^j A_j)$ and starts work only when the validation is passed. The Lagrange interpolation formula is used for more than n working nodes: $F(x) = \sum_{i=1}^{t}(y_i \prod_{i\leqslant j\leqslant t, j\neq i}(x - x_j)(\prod_{i\leqslant j\leqslant t, j\neq I}(x_j - x_i)^{-1})$, to restore the image storage hash. Since splits stored in an SC are encrypted, the worker node has received splits and needs to decrypt them using its private key. The DO has already finished retrieving the worker node's public key when the image is uploaded.

The end of the transaction is marked by the SU's obtaining the image from IPFS, and after decryption, it is verified. The access control downloaded from IPFS is explained in the previous section. Even after the transaction, violations can be defended through SCs, mainly for violations such as piracy. As described in the above section, SCs calculate the Tadors score to determine whether there is a violation.

5 Security Analysis

This section analyzes the security of the proposed solution and provides corresponding security explanations for possible threats.

(1) Single point of failure, single point of failure refers to the failure of single or multiple nodes in a distributed system affecting the operation of the system. The proposed scheme uses the PoW consensus algorithm for data synchronization, and a failure of less than half of the nodes will not affect the operation of the scheme.

(2) From the perspective of the data owner, although the blockchain is open and transparent, the data stored by the DO are encrypted. In the retrieval process, PPIR introduces SSE, eliminating the risk of image reconstruction. In the design of copyright protection, symmetrical fingerprints are used in the scheme, and homomorphic encryption is used to protect fingerprint information. The fingerprint information will only be revealed if DO launches a piracy complaint. If the DO does not want to reveal its identity, we can protect the privacy of the DO through the anonymity mechanism.

(3) From the perspective of the SU, our scheme uses SSE to ensure that the SU's search will not reveal the uses' preferences and that the blockchain will not collect and analyze information about users. In addition, our solution is designed to prevent frame-up (CWE and homomorphic encryption) because the DO cannot frame an innocent SU by constructing a fingerprint code.

6 Performance Evaluation

Experimental configuration is as follows: DELL workstation T7920, with 64 GB RAM, 500 MB bandwidth. Ten physical nodes were built using Raspberry PI to build the Ethereum network. We wrote the SC using Solidity and Javascript and tested the gas and time consumption of the SC. PPIR uses the public image dataset Caltech 256 and the ORL face database to test the retrieval performance, including the retrieval time and average precision (namely mAP). In the use of the CNN, resnet 50 is called to pretrain the model, and the obtained feature vector is 2048-dimensional. PCA can be used to encode and compress the feature vectors according to different scenarios.

6.1 Smart contract

This section mainly tests the performance of the SC used in our proposed scheme, including gas and time consumption. We mainly test SCs in four aspects: SSE-based retrieval, feature vector storage, copyright protection, and price game.

(1) Image retrieval

$index\text{-}store(\,)$ is called by the DO to store the index generated by the improved LSH. The index is stored in the SC as a data structure of the bucket table, and the different bucket values are uploaded as multitransaction $\#TX$. The $Search(\,)$ function is called by the SU to complete the query request. The process of image retrieval is to map the searched image to ALSH and then compare it with the hash value of each bucket table (the bucket table can be considered a linked list, and the comparison result of the head node determines whether to map the feature vector to the bucket table). The complexity of the search is determined by the number of bucket tables because every search is a comparison of buckets. The more bucket tables, the higher the complexity of the search. Although PPIR performs XOR on multiple hash values, it still generates more bucket tables in large-scale image retrieval. $Add(\,)$ and $Update(\,)$ are called by the DO to add and update the index. Each add and update requires a remapping of the feature vector and adjusting the token (add or remove) to output the final index to complete the function call.

(2) Image store

Three functions are used in the SC of image storage. $WNPK(\,)$ is used to obtain the public key of the working node, and then it is used to encrypt the hash value of segmentation. Next, the $Hash\text{-}store(\,)$ function is used to store the encrypted hash value. $SSS(\,)$ is used for secret reconstruction, and the working node calculates Lagrangian interpolation through the $SSS()$ function. When no less than n nodes complete the calculation, the address hash of the image can be reconstructed.

(3) Copyright preservation

An SC for copyright protection has two functions: $CW(\,)$ and $Tracking(\,)$. The $CW(\,)$ function is called in the trusted execution environment SGX to generate the collusion-resistant Tardos code and store the resulting fingerprint code in the SC. Similarly, $Tracking(\,)$ must be invoked from SGX and compared with the data on the chain, and the final score is output to identify and punish participants involved in piracy.

(4) Fair trading

Two main functions are used in the fair trading SC. $Bargain(\,)$ completes the bargaining process and outputs the final price. $Trading(\,)$ completes the verification of receipts and transaction parties, and provides validation results.

In terms of encoding the Ethereum SC, we used Ganache+Truffle to build the Ethereum test network and Solidity deployment. We mainly tested the performance of the following ten functions: $Bargain(\,)$, $Index\text{-}store(\,)$, $Search(\,)$, $Update(\,)$, $SSS(\,)$, $CW(\,)$, $Tracking(\,)$, $Trading(\,)$, $WNPK(\,)$, and $Hash\text{-}store(\,)$. Figure 4 shows the time consumption of each function. The two functions, $Index\text{-}store(\,)$ and $Search(\,)$ have high time consumption because they require more iterative calculation. In an SC, more iterations mean higher computational complexity, thus greater performance consumption. Since the $CW(\,)$ and $Tracking(\,)$ functions must be calculated several times, we calculate the average value and give the final result. In addition, the computational complexity of other functions is low, and the time consumption is within a reasonable range, which can be applied to large-scale image trading scenarios. We also tested the gas consumption of the function, and the result is shown in Fig. 5. Gas consumption is also related to the complexity, similar to the time consumption, so we will not describe it in detail.

6.2 Precision of image retrieval

PPIR uses a CNN and a pretrained model to extract image feature vectors and uses an improved LSH function to reduce the false negative rate. These effectively combined methods improve the accuracy of retrieval. We tested the accuracy of mAP and a search of the top-U images on the Caltech256 database and compared it with similar research work[16], and top-U represents the most similar U images. Figure 6 shows the result of the PPIR test of mAP. The higher the dimension of the feature vector, the higher the value of mAP. When the dimension is lower than 512, mAP will drop straight down, indicating that we should not choose a dimension that is too low, or the value of mAP will be reduced. Figure 7 shows the result of retrieving the top-U similar images. The retrieval accuracy of the improved LSH function is better than that of the traditional one. The accuracy of our scheme in mAP and top-U is better than that of existing similar schemes, which further shows that our proposed PPIR scheme is feasible.

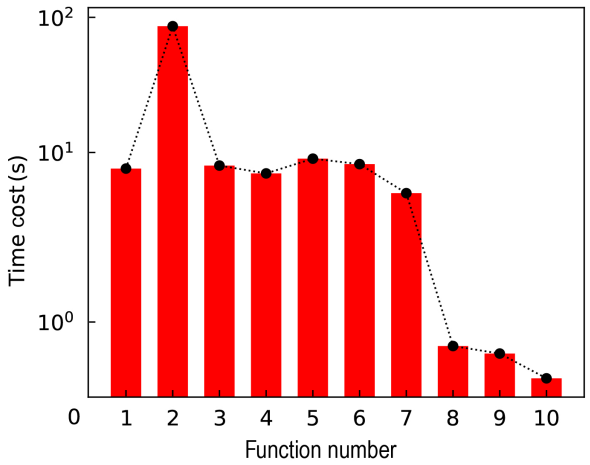

Fig. 4 Time cost of ten functions.

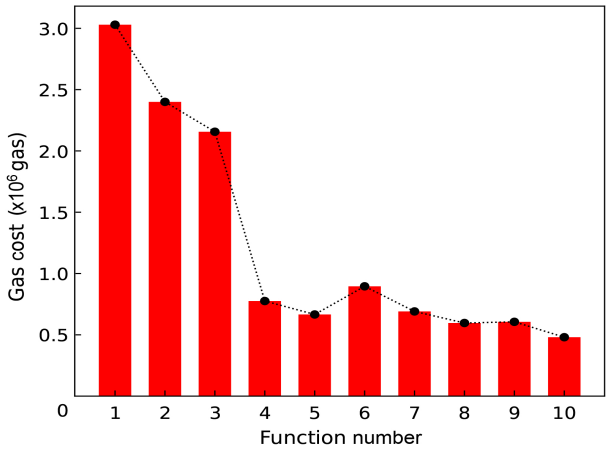

Fig. 5 Gas cost of ten functions.

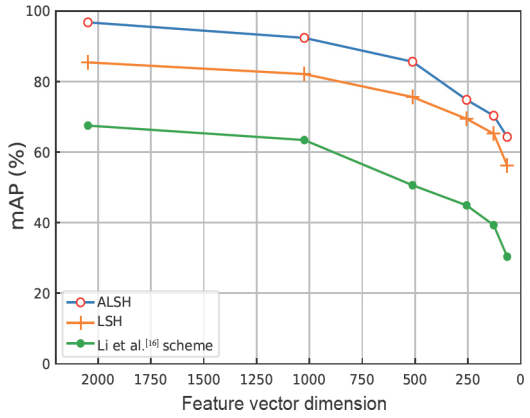

Fig. 6 mAP with different feature vector dimensions.

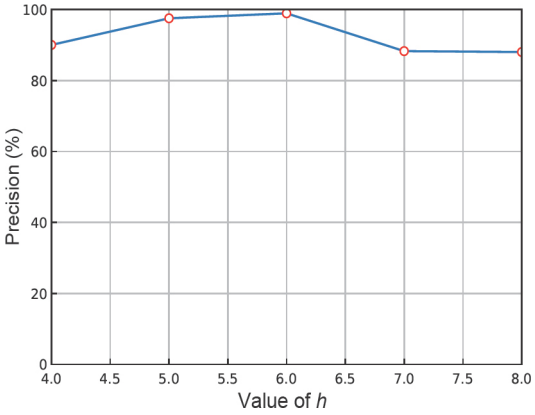

Fig. 8 Influence of different h values on precision.

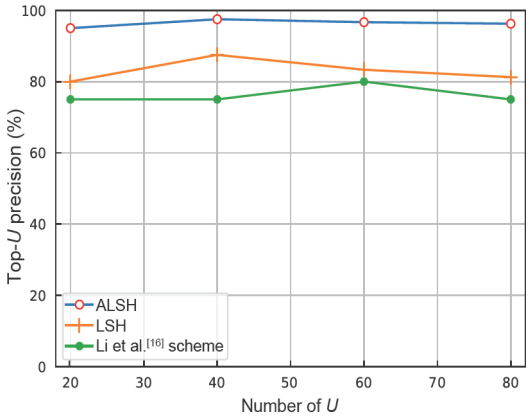

Fig. 7 Comparison of Top-U precision between similar jobs.

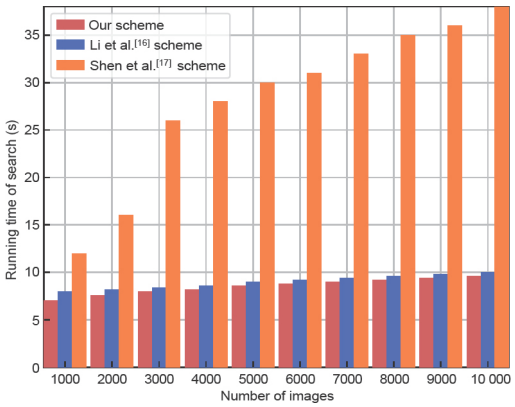

Fig. 9 Running time of image retrieval with different sizes with a feature vector dimension of 64.

6.3 Search time and efficiency of SSS and IPFS

We tested the influence of the value of k of the improved LSH function on the accuracy. We set the dimension as 2048, and the value of k ranged from 4 to 8. As shown in Fig. 8, the accuracy changes as h increases. The optimal value of h for the Caltech256 dataset is 6. To make a better comparison with the existing similar work[16], we tested the time consumption of different dataset sizes of the same dimension. Figure 9 shows that our scheme has low time consumption and can meet the requirements of large-scale image retrieval scenarios. In addition, we also tested the SSS split time. Figure 10 shows the average CPU time when the hash value split was performed. After testing, the CPU consumed the least time, 0.34 ms, when the number of partitions was 5 or 3. In addition, our scheme uses IPFS to store and download pictures instead of a cloud server. We tested the download speed of 0–5 GB images in Fig. 11, outperforming the cloud server.

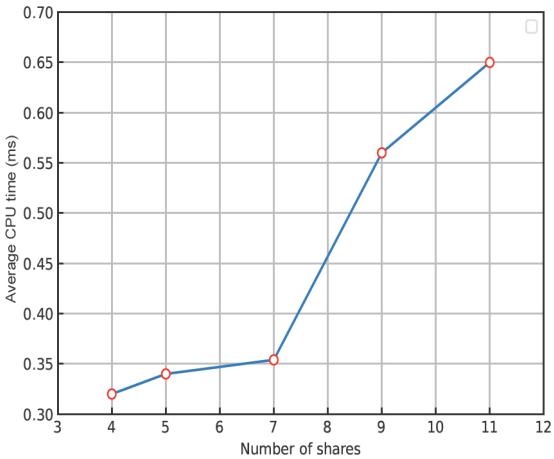

Fig. 10 Computational time versus number of shares.

6.4 Collusion of several buyers

Figure 12 shows the time cost of code generated and piracy detection against collusion attacks. The maximum number of collaborators was determined because Tardos was used, and we tested the time with 10–250 participants through simulation experiments. We used SCs for generation and detection, and the experiments show that the time cost is within a reasonable and acceptable range.

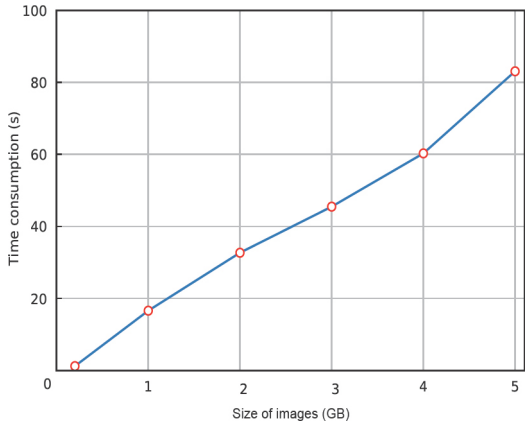

Fig. 11 Time cost of the download with IPFS.

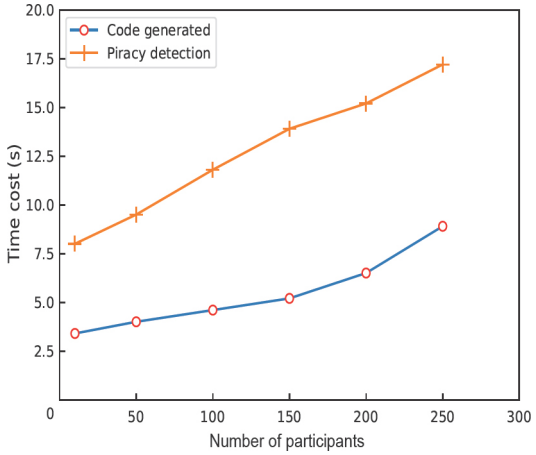

Fig. 12 Time cost of code generated and piracy detection.

7 Conclusion

We implemented a decentralized image trading scheme, considered all the processes in the transaction, and designed three models on blockchain for this trading scheme, namely, a privacy-preserving image retrieval scheme, a copyright-preserving scheme of images, and a fair image trading scheme. The experiment shows that the proposed scheme has certain advantages and outperforms other schemes. In the future, we will conduct more in-depth research on the security of the system and larger-scale applications. At the same time, we also have great interest in some of the latest technologies, such as the use of path verification to verify image retrieval results and the recently emerging copyright protection technology NFT. We will combine these new technologies for further research in future work.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62062016 and U21A20474), Guangxi Natural Science Foundation (No. 2019JJA170060), Jiangsu Provincial Key Laboratory of Network and Information Security (No. BM2003201), and Guangxi Science and Technology project (Nos. GuikeAA22067070 and GuikeAD21220114). Finally, we thank the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, Center for Applied Mathematics of Guangxi (Guangxi Normal University), and the Guangxi Talent Highland Project of Big Data Intelligence and Application.

References

[1] O. B. Sezer and A. M. Ozbayoglu, Financial trading model with stock bar chart image time series with deep convolutional neural networks, arXiv preprint arXiv:1903.04610, 2019.

[2] N. Cohen, T. Balch, and M. Veloso, Trading via image classification, in *Proceedings of the First ACM International Conference on AI in Finance ACM*, New York, NY, USA, 2020. pp. 1–6.

[3] W. Sae-Tang, M. Fujiyoshi, and H. Kiya, A new copyright- and privacy-protected image trading system using a novel steganographybased visual encryption scheme, *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, vol. 17, no. 1, pp. 95–107, 2019.

[4] W. Sae-Tang and H. Kiya, Hadamard transform-based amplitude-only images for image trading systems, in *Proc. of International Workshop on Advanced Image Technology*, Busan, Republic of Korea, 2016, pp. 1–4.

[5] Z. Ma, Digital rights management: Model, technology and application, *China Communications*, vol. 14, no. 6, pp. 156–167, 2017.

[6] T. Chen, T. C. Chang, and T. Zhu, Security-enhanced cloud-based image secret sharing and authentication using POB number system, *Multimed. Tools Appl.*, vol. 80, no. 2, pp. 1901–1924, 2021.

[7] Z. Wu, H. Li, X. Wang, Z. Wu, L. Zou, L. Xu, and M. Tan, New benchmark for household garbage image recognition, *Tsinghua Science and Technology*, vol. 27, no. 5, pp. 793–803, 2022.

[8] L. Wang, X. Liu, and X. Lin, A fair and privacy-preserving image trading system based on blockchain and group signature, *Secur. Commun. Netw.*, vol. 2021, p. 5701258, 2021.

[9] J. Song, E. Kang, H. W. Shin, and J. Jang, A smart contract-based P2P energy trading system with dynamic pricing on ethereum blockchain, *Sensors*, vol. 21, no. 6, pp. 1985, 2021.

[10] T. Li, H. Wang, and D. He, Blockchain-based privacypreserving and rewarding private data sharing for IoT, *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15138–15149, 2022.

[11] J. Shen, Blockchain technology and its applications in digital content copyright protection, in *Proc. 4^{th} Int. Conf. Economic Management and Green Development*, Singapore, 2021, pp. 18–25.

[12] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, *Decentralized Business Review*, vol. 21, no. 5, pp. 21260–21268, 2008.
[13] G. B. Mermer, E. Zeydan, and S. S. Arslan, An overview of blockchain technologies: Principles, opportunities and challenges, in *Proc. 2018 26th Signal Processing and Communications Applications Conf.* (*SIU*), Izmir, Turkey, 2018, pp. 1–4.
[14] J. Ren, J. Li, H. Liu, and T. Qin, Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT, *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 760–776, 2022.
[15] H. Ma, H. Ding, Y. Yang, Z. Mi, J. Y. Yang, and Z. Xiong, Bayes-based arp attack detection algorithm for cloud centers, *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 17–28, 2016.
[16] X. Li, J. Li, F. Yu, J. Yang, and Y. Chen, BEIR: A blockchain-based encrypted image retrieval scheme, in *Proc. 2021 IEEE 24th Int. Conf. Computer Supported Cooperative Work in Design* (*CSCWD*), Dalian, China, 2021, pp. 452–457.
[17] M. Shen, Y. Deng, L. Zhu, X. Du, and N. Guizani, Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach, *IEEE Netw.*, vol. 33, no. 5, pp. 27–33, 2019.
[18] M. Li, M. Zhang, Q. Wang, S. S. M. Chow, M. Du, Y. Chen, and C. Lit, InstantCryptoGram: Secure image retrieval service, in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Computer Communications*, Honolulu, HI, USA, 2018, pp. 2222–2230.
[19] S. Hu, C. Cai, Q. Wang, C. Wang, Z. Wang, and D. Ye, Augmenting encrypted search: A decentralized service realization with enforced execution, *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 6, pp. 2569–2581, 2019.
[20] A. Savelyev, Copyright in the blockchain era: Promises and challenges, *Comput. Law Secur. Rev.*, vol. 34, no. 3, pp. 550–561, 2018.
[21] S. Tulyakov, F. Farooq, and V. Govindaraju, Symmetric hash functions for fingerprint minutiae, in *Proc. Third Int. Conf. Pattern Recognition and Image Analysis*, Bath, UK, 2005, pp. 30–38.
[22] A. Charpentier, C. Fontaine, T. Furon, and I. Cox, An asymmetric fingerprinting scheme based on Tardos codes, in *Proc. 13th Int. Workshop on Information Hiding*, Prague, Czech Republic, 2011, pp. 43–58.
[23] F. Farooq, R. M. Bolle, T. Y. Jea, and N. Ratha, Anonymous and revocable fingerprint recognition, in *Proc. 2007 IEEE Conf. on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, 2007, pp. 1–7.
[24] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, Digital rights management for content distribution, in *Proc. Australasian Information Security Workshop Conf. ACSW Frontiers 2003*, Adelaide, Australia, 2003, pp. 49–58.
[25] Z. Ma, M. Jiang, H. Gao, and Z. Wang, Blockchain for digital rights management, *Fut. Generat. Comput. Syst.*, vol. 89, pp. 746–764, 2018.
[26] S. Kenny and L. Korba, Applying digital rights management systems to privacy rights management, *Comput. Secur.*, vol. 21, no. 7, pp. 648–664, 2002.
[27] C. Huang, D. Liu, J. Ni, R. Lu, and X. Shen, Achieving accountable and efficient data sharing in industrial internet of things, *IEEE Trans. Ind. Inf.*, vol. 17, no. 2, pp. 1416–1427, 2021.
[28] Z. Chen, Y. Wang, and T. Ni, DCDChain: A credible architecture of digital copyright detection based on blockchain, arXiv preprint arXiv: 2010.01235v2, 2020.
[29] D. Sheng, M. Xiao, A. Liu, X. Zou, B. An, and S. Zhang, CPchain: A copyright-preserving crowdsourcing data trading framework based on blockchain, in *Proc. 2020 29th International Conference on Computer Communications and Networks* (*ICCCN*), Honolulu, HI, USA, 2020, pp. 1–9.
[30] Y. Wei and C. Jin, Locality sensitive discriminant projection for feature extraction and face recognition, *J. Electr. Imag.*, vol. 28, no. 4, p. 043028, 2019.
[31] Q. Wang, X. Zhang, J. Qin, J. Ma, and X. Huang, A verifiable symmetric searchable encryption scheme based on the AVL tree, *Comput. J.*, vol. 66, no. 1, pp. 174–183, 2023.
[32] M. A. Saviour and D. Samiappan, IPFS based file storage access control and authentication model for secure data transfer using block chain technique, *Concurr. Comput. Pract. Exp.*, vol. 35, no. 2, p. e7485, 2023.
[33] A. Rubinstein, Perfect equilibrium in a bargaining model, *Econometrica*, vol. 50, no. 1, pp. 97–109, 1982.
[34] F. Radenović, G. Tolias, and O. Chum, CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples, in *Proc. 14th European Conf. Computer Vision*, Amsterdam, the Netherlands, 2016, pp. 3–20.
[35] N. Ren, D. Tong, H. Cui, C. Zhu, and Q. Zhou, Congruence and geometric feature-based commutative encryption-watermarking method for vector maps, *Comput. Geosci.*, vol. 159, p. 105009, 2022.
[36] B. Škorić and J. J. Oosterwijk, Binary and q-ary Tardos codes, revisited, *Des. Codes Cryptogr.*, vol. 74, no. 1, pp. 75–111, 2015.

Jiahui Peng received the MEng degree from Guangxi Normal University, China in 2022. He is working at Guangxi Power Grid Materials Co Ltd. His research interests include blockchain systems, data security, and Internet of Things. He was awarded the Best Applied Paper at the 9th International Conference on Advanced Cloud and Big Data.

Feng Yu received the PhD degree from Southeast University, China in 2015. She is currently an associate professor at Guangxi Normal University, China. Her main research fields are data distributed system, data security, blockchain, and cloud computing.

Xianxian Li received the PhD degree from Beihang University, China in 2002. He is currently a professor at the School of Computer Science & Engineering, Guangxi Normal University, China. His research interests include data security, distributed system security, Internet of Things, and software theory. He has published over 60 refereed papers in these areas. He has served as a program co-chair/technical program committee member for several IEEE conferences and workshops.

Chunpei Li received the MEng degree from Guangxi Normal University, China in 2020. He is currently a PhD candidate in software engineering at Guangxi Normal University, China. His research interests include blockchain systems, data security, and Internet of Things.

Bin Qu received the MEng degree from Guangxi Normal University, China in 2020. He is currently at the School of Computer Science & Engineering, Guangxi Normal University, China. His research interests include edge computing, reinforcement learning, Internet of Things, and blockchain system.