DCDChain: A Credible Architecture of Digital Copyright Detection Based on Blockchain ⋆

Zhili Chen[a,b], Yuting Wang[a,b], Tianjiao Ni[a,b], Hong Zhong[a,b]

[a] *School of Computer Science and Technology Anhui University Hefei, China*
[b] *Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China*

Abstract

Copyright detection is an effective method to prevent piracy. However, untrustworthy detection parties may lead to falsified detection results. Due to its credibility and tamper resistance, blockchain has been applied to copyright protection. Previous works mainly utilized blockchain for reliable copyright information storage or copyrighted digital media trading. As far as we know, the problem of credible copyright detection has not been addressed. In this paper, we propose a credible copyright detection architecture based on the blockchain, called DCDChain. In this architecture, the detection agency first detects copyrights off the chain, then uploads the detection records to the blockchain. Since data on the blockchain are publicly accessible, media providers can verify the correctness of the copyright detection, and appeal to a smart contract if there is any dissent. The smart contract then arbitrates the disputes by verifying the correctness of detection on the chain. The detect-verify-and-arbitrate mechanism guarantees the credibility of copyright detection. Privacy and security analysis and experimental simulations show that the digital copyright detection architecture is reliable, secure and efficient.

Keywords: `blockchain`, copyright detection, smart contract

⋆Fully documented templates are available in the elsarticle package on CTAN.

Email addresses: `zlchen@ahu.edu.cn` (Zhili Chen), `yt_wang11@163.com` (Yuting Wang), `tjni94@163.com` (Tianjiao Ni), `zhongh@mail.ustc.edu.cn` (Hong Zhong)

1. Introduction

With the popularization of digital technologies, there are more and more ways to copy and transmit digital media [1]. Since many netizens lack a good awareness of copyright problems, copyrights are often not fully protected, and the benefits of owners are damaged frequently. To alleviate the copyright problems, researches on copyright protection are crucial.

Recently, a large number of papers have proposed some measures to prevent piracy [2]. A common approach is plagiarism detection which is mainly to determine whether the media has copied or plagiarized content. Paper [3] proposes that texts are segmented and then tested. And plagiarism detection in text using the vector space model[4]. Change et al.[5] first present the content-based image plagiarism detection. Then to resist arbitrary rotation, an efficient image copy detection [6] is proposed. Wary et al.[7] review the research progress of piracy detection in video. These researches depend on the detection agencies or detection tools which maybe not fully credible. If the detection agencies are corrupted or the detection tools [8] (such as Parikshak, Copy Scape) crashes, the loss will be immeasurable. The research [9] also proves that the detection tools may not get correct detection results. So plagiarism detection tools should not be blindly trusted.

Due to the credibility and tamper resistance nature, blockchain has received extensive attention in the copyright protection field [10, 11, 12, 13, 14]. However, it is only used as a trading platform[15] to earn copyright fees for copyright owners [10, 16], or to reliably store media copyright records [11, 12, 13, 14]. In [17, 18], an image is embedded with digital watermarks, and its hash value is stored on the blockchain, ensuring that the existence proof of the exact copyrighted image cannot be forged. As far as we know, there is a small amount of blockchain-based work to address the issue of copyright detection, where media content may be completely or partially pirated. The smart contracts that use perceptual hashing and ethereum automatically detect and reject tampered images that are perceptually similar to those already on the market [19]. But the design is only for the image, the lack of different media piracy detection consideration. There is no mention of the full design process of the smart contract and the security considerations for the possibility of copyright theft of the image itself.

In this paper, we propose a credible copyright detection architecture. Specifically, to ensure reliable detection results, the detection agency detects the copyrights of media locally, and stores the results on blockchain. Then due to the public accessibility of blockchain, the media providers can verify the correctness of the detection results locally, and appeal to a smart contract for arbitration if there is any dissent. The smart contract then arbitrates the disputes by the detection results on the chain. Furthermore, financial incentives are also used to stimulate both the detection agency and the media providers. Our main contributions can be summarized as follows:

- To our best knowledge, we first propose a common copyright detection scheme based on blockchain, which ensures credible copyright detection and is suitable for various media.
- By combining secure hash algorithm and locality sensitive hashing innovatively, our scheme performs efficient detection.
- The detect-verify-and-arbitrate mechanism guarantees the credibility and accuracy of copyright detection.
- On the one hand, we conducted experiments in four different corpora and used regression models to determine the relationship between hamming distance and similarity. On the other hand, we implement smart contract on Rinkeby testnet and compare with local Ganache in term of time. Experimental results show that our scheme is feasible and efficient.

The rest of the paper is organized as follows. Section 2 gives technical preliminaries. Section 3 describes the situation analysis and design targets. In Section 4, we present the overview and the detailed design of our scheme. Section 5 is evaluation and discussion. Finally, Section 6 concludes our work.

2. TECHNICAL PRELIMINARIES

In this section, we introduce some technologies related to our architecture.

2.1. Blockchain

Blockchain [20, 21] is a decentralized data ledger shared by all nodes. Except for the genesis block, each block contains the hash value of the previous block and a timestamp indicating write time of data, so any data on blockchain are tamper-resistant. The node encapsulates transactions into a block and links to the current longest main blockchain. Every transaction can be traced through this chain structure. Also, every transaction in the block contains the initiator's digital signature to ensure authenticity and legality. The tamper resistance, traceability, and authenticity of the blockchain are suitable for copyright detection.

Smart contracts are event-driven computer programs that are deployed in the blockchain. They are regarded as special transactions that are easily traced and can not be tampered or reversed. Once the predefined conditions are met, the code on the contract can be executed automatically without the external authorization. Considering the advantages of blockchain, we design a smart contract for copyright detection.

2.2. Hash algorithm

Our scheme is based on two hash algorithms: the secure hash algorithm and locality sensitive hashing. The marked hash value of digital media is implemented by the secure hash algorithm.

Definition 1 (Secure Hash Algorithm) A one-way hash function is a function H which accepts an arbitrarily large input x, and produces a small fixer-size output h: $h = H(x)$. It is a function with the following basic properties.

- Collision resistance: it should be difficult to find distinct input x,x' such that $H(x) = H(x')$.

- Determinism: it is a data conversion function that maps inputs (raw data) to fixed-length outputs (indexes).

We calculate the media similarity based on the locality sensitive hashing.

Definition 2 (Locality Sensitive Hashing)[22] For any two points q, v in space S, the distance $D(q, v)$ of two points such as Euclidean distance, Manhattan distance, etc., if the function $H = \{h : S \to U\}$ meets the following two conditions:

(1) $D(q, v) \leq r, Pr_H[h(q) = h(v)] \geq p_1$

(2) $D(q, v) > r(1 + \varepsilon),\ Pr_H[h(q) = h(v)] \leq p_2$

Where ε, r are positive integers and $p_1 > p_2$, $H = \{h : S \to U\}$ is locality sensitive hashing.

2.3. IPFS

The Interstellar File System (IPFS) is a peer-to-peer distributed file system that aims to replace HTTP. IPFS combines many good ideas in the current file systems. BitSwap Protocol solves the problem of file sharing. Merkle DAG is another important part of IPFS because it guarantees that IPFS has useful functions such as content addressing, tamper resistance and deduplication. It replaces the domain-based address with a content-based address and the user can find the file stored in it by the address hash value.

3. SITUATION ANALYSIS AND DESIGN TARGETS

3.1. Situation Analysis

In the real copyright detection process, there may be opacity of the detection process and the evaluation criteria. For example, for the same paper, different paper examination tools will have different reproduction ratios. So a paper will have different test results, and we do not know who can be believed. Second, the tester may maliciously provide incorrect test results. The mistakes of this result are not easy to find. Even if we suspect the test results, we cannot submit our comments.

3.2. Design Targets

Our design achieves the following three targets.

Tamper resistance: All copyright data should be stored tamper-resistantly, such that anyone cannot change the copyright data maliciously. Blockchain provides a tamper-resistant approach for copyright data storage.

Credibility: The copyright detection results should be entirely reliable. The detection process should be carried out in a credible way, and the detection results should be publicly verifiable.

Efficiency: The copyright detection should be sufficiently efficient. It should be fast enough to detect the copyrights of some media products.

4. DCDCHAIN

In this section, we propose a credible architecture of digital copyright detection based on blockchian, called DCDChain.

4.1. Overview

In order to achieve the above design targets, we adopt the following methods to design the architecture.

Blockchain: We assume that the blockchain holds all the copyright-protected material in a certain domain. Furthermore, to automatically and reliably arbitrate the detection results, we design a smart contract that has arbitration and auto-transfer functions. Once the smart contract arbitrates that the submitted results by detection agency are wrong, the transfer function that sends ether to media providers will be executed automatically.

Hash algorithms: It is unrealistic to store media directly on the blockchain. A more practical approach [11, 23] is storing the hash value uniquely indexing media. the value is calculated by the SHA256. Once the digital media have any change, the hash value changes significantly. Furthermore, since the direct copyright detection with media contents is very computation-intensive, we adopt the locality sensitive hashing (LSH) to calculate the similarities of digital media, and it can improve the detection efficiency.

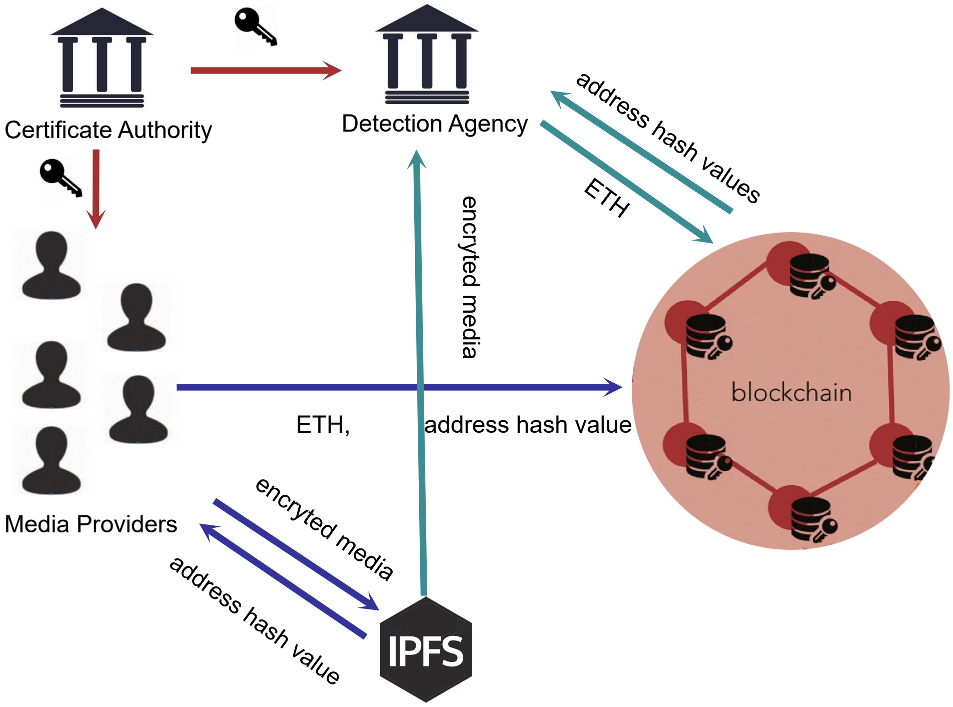


Figure 1: Overview of the architecture

As illustrated in Figure 1, the five entities comprising our architecture are Detection Agency, Media Providers, Certificate Authority, Blockchain, and IPFS.

Detection Agent (DA): The purpose of DA is to earn service fees through detection services ordered by media providers. Before this, the DA first pays the smart contract a deposit. Detection services include the detection of pirated media and the announcement of results according to encrypted media. Its account and identity are public. It is worth noting that the detection algorithm also needs to be made public. If the identity or algorithm is not disclosed by the testing organization, the media provider may not pay the service fee. Therefore, if DA is dishonesty, its reputation will be greatly damaged. In addition, the public detection algorithm plays an important role in media providers' verification detection results.

Media Providers (MPs): MPs aim to obtain credible copyright detection services. If they want to obtain detection services, they must pay the smart contract service fees and send encrypted media to IPFS. Then it sends the address hash value to the smart contract. MPs have the ability to verify detection results. If they suspect the results, they are able to issue a doubt to the smart contract.

Blockchain: The stored address hash value by MPs is open and trustworthy in blockchain. DA deploys the smart contract with arbitration and transfer functions on the blockchain. The arbitration function includes two aspects: one is to determine whether hash values provided by MPs correspond to legal media; the other is to check whether detection results of doubtful content satisfy the piracy confirmation rule.

Certificate Authority (CA): After authenticating the applicants (DA and MPs), CA sends them the digital certificates for their public keys.

Inter Planetary File System (IPFS): For the media storage, IPFS stores the complete media contents while blockchain saves the hash values. It returns address hash value to MPs.

The main notations of the architecture are shown in Table1.

4.2. Detail Design

We apply a hybrid approach that incorporates blockchain, hash algorithms and IPFS into our design. It allows DA to calculate the similarities of media credibly, while the public is able to trace copyright information. As the trusted party, CA authenticates the identities of DA and MPs, and issues the digital certificates including public keys PK_{DA}, PK_i to them, respectively. The corresponding private keys SK_{DA}, SK_i are kept secret individually.

The proposed credible and tamper resistance media transaction architecture based on blockchain model consists of three stages: preliminary stage, detect-verify-arbitrate stage, modify history stage. The proposed architecture is depicted in Figure 1.

Table 1: NOTATIONS

Notations	Descriptions
m	Medium to be detected
$hashID$	The result of calculation of the SHA256 of m
$lshv$	The result of calculation of the LSH of m
lm_j	The j-th legal media
$hashID_j$	The result of calculation of the SHA256 of lm_j
$lshv_j$	The result of calculation of the LSH of lm_j
f_i	The service fee
f_{DA}	The deposit
L	Hamming distance
θ	Threshold of hamming distance

4.2.1. Preliminary Stage

There are three steps in the preliminary stage. First, the media provider MP_i and the detection agent DA authenticate each other with digital certificates released by the certificate authority. Second, the original media file m is encrypted with mix encryption to ensure that only the detection agent DA can decrypt it. Third, the encrypted media file is uploaded to IPFS and its hash address is handed to the smart contract and thus also to the detection agent DA. At the same time, the media provider MP_i commits a testing service fee and the detection agent DA commits a security deposit to the smart contract. The detailed description is as follows.

Step 1: The media provide MP_i and the detection agent DA verify each other's identity by digital certificates as follow:

(1) MP_i sends its digital certificate C_{MP_i} to DA, and DA sends its digital certificate C_{DA} to MP_i.

(2) Both MP_i and DA verify each other's digital certificates with the public key of the certificate authority.

(3) If either MP_i or DA finds an invalid digital certificate, it abort the protocol.

Otherwise, both of them extract the identity and public key of each other from the digital certificate.

Step 2: The media provider MP_i encrypts the media file m with the mix encryption. Specifically, MP_i encrypts the media by asymmetric encryption E^a and symmetric encryption E^s as eq. (1) and gets the cipertext S_i^m.

$$S_i^m = (E^a_{PK_{DA}}(k_i), E^s_{k_i}(m)) \tag{1}$$

which can be decrypted only by the detection agent DA with its private key SK_{DA} as eq. (2).

$$\begin{aligned} k_i &= D^a_{SK_{DA}}(E^a_{PK_{DA}}(k_i)) \\ m &= D^s_{k_i}(E^s_{k_i}(m)) \end{aligned} \tag{2}$$

Step 3: The media provider MP_i uploads S_i^m to IPFS and gets the address hash value Q_i^m. MP_i then stores Q_i^m in the smart contract denoted by SC, which has been deployed on the blockchain by the detection agent DA. At the same time, MP_i sends a service fee f_i and DA sends a deposit f_{DA} to SC. The fund commitments are used to penalize the party who perform dishonestly, which will be seen later.

The preliminary stage above guarantees that the corresponding S_i^m can be found through Q_i^m and only DA can decrypt it to get the original content of m. However, there is only read permission for DA, and it cannot tamper with the contents of m. The existing record of Q_i^m in IPFS can be used as an important proof of existence for copyright protection of m.

4.2.2. Detect-Verify-Arbitrate Stage

In this stage, the detection agent DA first obtains the content of the media file m, calculates its hash values, and detects if this media file is pirated or not. This detection result will be verified by media providers MP_i, and finally arbitrated by the smart contract SC. The work flow is shown as in Fig. 2.

Detect: The detection agent DA downloads S_i^m from IPFS by Q_i^m, and then decrypts S_i^m by SK_{DA} to get m. DA calculates $hashID$ and $lshv$ by SHA256 and LSH algorithms, respectively. At the same time, it downloads all hash values

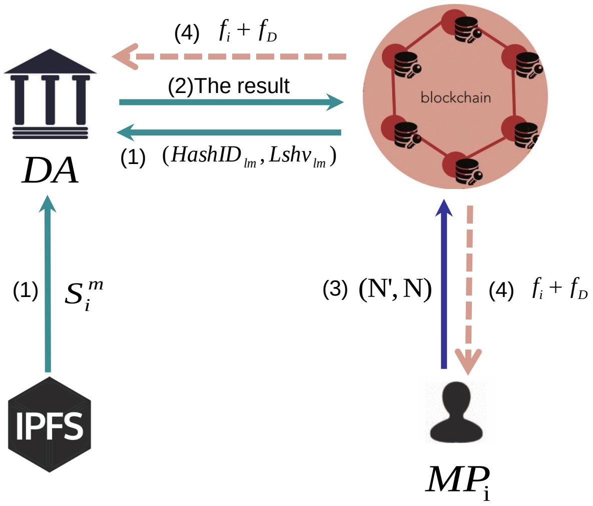


Figure 2: Detect-Verify-Arbitrate workflow

for legal media $HashID_{lm}$ and $Lshv_{lm}$ from the blockchain, where $HashID_{lm} = \{hashID_1, ..., hashID_j, ...\}$, $Lshv_{lm} = \{lshv_1, ..., lshv_j, ...\}$. Then,

(1) Match $hashID$ in $HashID_{lm}$, and if $hashID = hashID_j$ for some j, m is complete pirated media.

(2) If $hashID \notin HashID_{lm}$, calculate hamming distance $L(lshv, lshv_j)$ for all $lshv_j \in Lshv_{lm}$. If $L(lshv, lshv_j) <= \theta$ exists, m is partial pirated media.

Algorithm 1 gives the judgment rules for pirated media.

Note that in practical implementations, the detection agent DA can achieve more efficient searches as follows. DA maintains a local database exactly the same as that of the blockchain but with more efficient search structures, and detects media files with the local database and uploads detection results to the blockchain accordingly. DA can backup all detection results to its local database and only need to update the database when there is a new detection result.

The pirated results include two types: partial piracy and complete piracy.

Algorithm 1 The judgment rules for pirated media

```
Input: $m$
Output: the result
DA calculates $hashID, lshv$.
DA downloads $HashID_{lm}, Lshv_{lm}$.
if $hashID \in HashID_{lm}$ then
    Complete piracy.
else
    if $L(lshv, lshv_j) \leq \theta$ for some $lshv_j \in Lshv_{lm}$ then
        Partial piracy.
    else
        Legitimate media.
    end if
end if
DA sends the result to $SC$.
```

DA uploads the result, $[N, lshv, QM]$ or $[N, hashID, QM]$, to SC. Among them, N stands for the serial number of legitimate media on the blockchain detected by DA. QM represents the corresponding address hash value of the detected media on IPFS, which can be indexed to the original encrypted media. The address of the storage media and the address hash value correspond to each other, forming an untamable record on the blockchain. If there is a conflict, these can be used as proof of existence. In addition to the above, if m is detected as a legitimate media, DA uploads $\{N', hashID, lshv, QM\}$ to SC. Where N' is a temporarily stored legitimate media serial number. In order to ensure the correctness of the test results, next we need to verify-and-arbitrate the detect results.

Verify-and-Arbitrate: In order to make the detection result real and trustworthy, we use the economic incentive mechanism of the blockchain in our design, and the party with the correct detection result will get an amount of Ether. The source of the bonus is the deposit f_{DA} of DA, and the service fee

f_i of MP_i.

The smart contract SC is an important part of the blockchain. Since smart contracts are trusted and auto-executable, so we let SC take on the responsibility of validation and arbitration in the design. If the detect result is partial piracy or complete piracy. After DA uploads the result, SC obtains $hashID_j$ or $lshv_j$ of the legitimate media from N by mapping ($key => value$). SC uses the verification mechanism to directly verify the hash values to be detected according to the judgment rule. After successful verification, SC transfers the service fee f_i of this detection to DA. The transfer amount is set in advance. If only one detection is performed, the deposit f_{DA} is transferred to DA together.

If the detect result is legal, MP_i can verify the correctness of the copyright detection. If someone in MPs challenges the detect result, it can initiate arbitration claim to SC by providing supporting evidence $\{N', N\}$. Specifically, MP_i verifies the legitimate detect results submitted by DA to the blockchain locally, and then uploades supporting information to the blockchain. Because it can back up all the hash values to its own database, and only needs to update the database when new detect results are available. Within time T, SC obtains the newly detected hash values $\{hashID', lshv'\}$ and the hash values of the legal media $\{hashID, lshv\}$ from the support information $\{N', N\}$ by mapping ($key => value$), respectively. The SC conducts arbitration according to the piracy judgment rules. If MP_i challenges success, SC rewards MP_i with an ether. Set the reward amount in advance. If only one detection is performed, f_{DA} and f_i will be transferred to MP_i together. If MP_i is unsuccessful, it indicates that MP_i has verified incorrectly, and SC does nothing. If the time T is exceeded, it will time out. By default, all MPs agree with the detect result of DA, and SC will transfer the amount fee for this detect to DA. Set the transfer amount in advance. If only one detect is performed, the service fee f_i and the deposit fee f_{DA} will be transferred to the DA together. At the same time, the temporary legal detection result of DA is uploaded to the legal hash value database of the blockchain. The above is all the detection process.

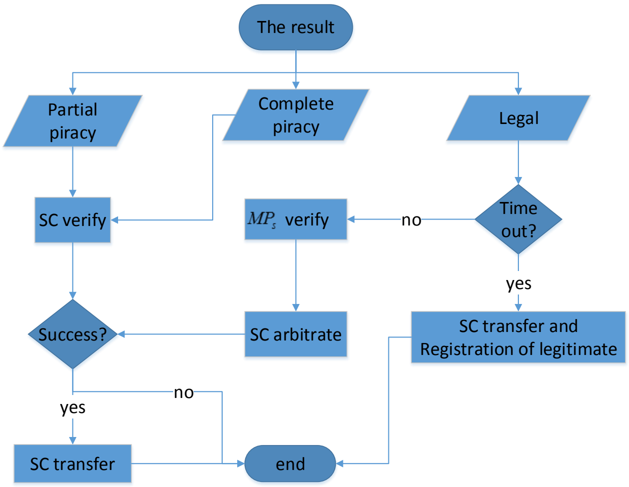


Figure 3: Verify-Arbitrate workflow

4.3. Privacy and Security Analysis

We rely on the blockchain being tamper resistance, an assumption that requires a sufficiently large and untrusted peer-to-peer network. In addition, we assume that there are enough legitimate hash values stored on the blockchain, which is the accumulation process of the legitimate media library. We now show how our architecture against adversaries.

In this architecture, only the user can control her data. The decentralized nature of blockchain and digitally-signed methods ensure that an adversary cannot impersonate a user or disrupt the network, as this means that the adversary has forged digital signatures or gained control over most of the network's resources. Similarly, the opponent cannot obtain any information from the public ledger because only the hash pointer is stored in it. S_i^m is stored in IPFS. The data of IPFS are tamper resistance and only the party that possesses the PK_{DA} can decrypt S_i^m.

If MP_i wants to get the detection service but does not pay f_i. SC pays f_i

after the detection result is correct, this process can be executed automatically without authorization of MP_i. It is unrealistic that DA wants to get f_i but does not responsible for detection results. First, MP_i will verify the detection result submitted by DA. Secondly, blockchain is transparent, if MP_i has doubts about the detection result, it lodges an appeal to SC by providing supporting information. SC can trace back to all transaction information according to transaction number on the blockchain and uses the above credible evidence to arbitrate. The economic incentive mechanism that reward the DA and MP_i with a certain amount of ether. Both parties have sufficient motivation to ensure the correctness of the detection results.

5. EVALUATION AND DISCUSSION

In this section, we demonstrate how DCDChain is reliable to implement an experimental simulation. All the experiments are performed on a machine with Intel(R) Core(TM) i5-6200U CPU @ 3.20 GHz and 8GB RAM, running Windows 10. The software used in the experiment includes Pycharm, Atom and Ethereum wallet(MetaMask).

5.1. Experimental simulation

DCDChain is a general architecture, but in this part, we use text for simulation.

5.1.1. Detection Section

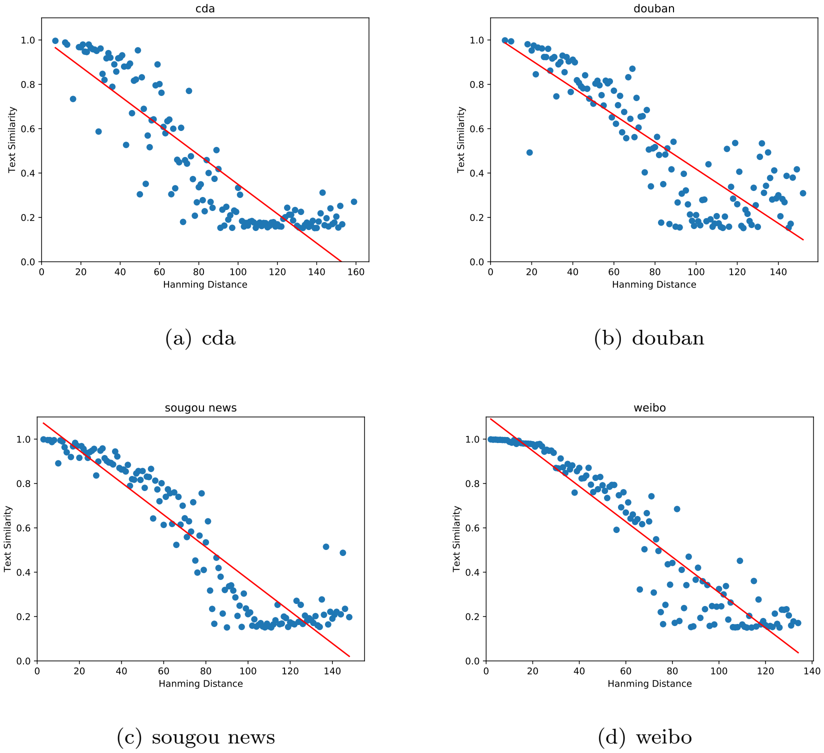
In LSH, there are several classic categories. The common categories in the text are k-shingle, simhash and minhash. The common categories in the image are perceptual hashing, average hash, and different hash algorithms. Our design aims to detect piracy in a large number of text, so we choose simhash for simulation.

We elaborate on the relationship between hamming distance and text similarity. The text determined to be pirated, the specific value of its threshold

θ. The experiment improved simhash. When reducing the dimensions of a keyword, the algorithm we use is SHA256. To discover the relationship between hamming distance and text similarity, we conducted experiments on multiple corpora and constructed regression models. The four corpora selected for the experiment are clinical document architecture(cda), douban multi-round dialogue(douban), sougou news, sina weibo(weibo) in Figure 4.

The cda corpus is a corpus composed of chief complaints and current medical history in a hospital's production environment. The length of the complaint is about 20 characters. The current medical history is about 100 characters and a total of 700,000 electronic medical records. The douban corpus is a public corpus, the corpus is completely segmented, the average text length is about 81 characters, and the corpus is a total of 520,000 dialogue records. The sogou news corpus is the news data generated by 18 channels of domestic, international, sports, social, entertainment, etc. from June to July 2012, taking the body part of the news, a total of 220,000 pieces of data, the average text length is about 210 character. The weibo corpus is a total of 410,000 microblogs generated between June and September 2018, with an average text length of approximately 90 characters. Based on these corpora, we select 1000 paragraphs of text, and randomly select 500 paragraphs of text for data enhancement to form new text. 1500 paragraph text is the sample point of our experiment. The specific data enhancement method is a random addition, deletion, and modification of the text.

The abscissa of the four figures represents the hamming distance and the ordinate represents the text similarity in Figure 4. The red line is a linear regression model constructed from sample points. According to the linear regression model, we can predict the corresponding similarity given the hamming distance. Another way of saying that given the threshold of text piracy we can predict the corresponding hamming distance.



(a) cda

(b) douban

(c) sougou news

(d) weibo

Figure 4: The relationship between Hanming Distance and Text Similarity

5.1.2. Validation and arbitration Section

To facilitate the demonstration, we write a smart contract that meets validation and arbitration requirements. React project was provided by the truffle framework, which assists us to accomplish the design of SC on the Rinkey testnet of the Ethereum. The block information of SC is shown in Table2. As a special transaction, SC will form a transaction when it is deployed successfully. SC contains two functions: $hashIDJudge()$ and $lshvJudge()$, and they will form the transactions once they are executed. These transactions have transaction number $transactionHash$, transaction index $txIndex$, block number $blockNumber$ and consumed gas value $gasUsed$. The $hashIDJudge()$ determines whether $hashID = hashID_j$. The $lshvJudge()$ determines whether $L(lshv, lshv_j) <= \theta$. SC transfers ether to the account with correct value if conditions are met.

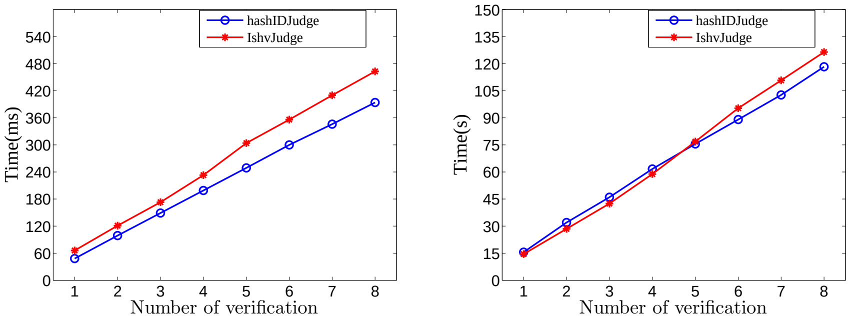
5.2. Performance and Overhead

We implement the scheme in Local Ganache and Rinkeby Testnet. Our aim is to study whether the detection process can be implemented and compare the time required on the local and test chain. In default, we set only one detection task.

We observe the efficiency of arbitration functions on the local Ganache and the Rinkeby testnet. As shown in Figure 5, the abscissa indicates the number of arbitration the supporting information, which are provided by a single user(MP_i). The ordinate indicates execution time. The blue line represents $hashIDJudge()$ and the red line represents $lshvJudge()$. In Figure 3(a), execution time that SC arbitrate one result fluctuates around 60 ms. In Figure 3(b), the execution time of SC arbitrate one result on the Rinkeby testnet fluctuates around 15s. The execution time on the Rinkeby testnet was significantly higher than on the local Ganache. For the different detection results, the execution time of the validation function is slightly different. In general, the time varies linearly with the number. As the number increases, so does the execution time. Only when users are honest in their detection or verification of copyright can

Table 2: The block information of *SC* on the Rinkeby testnet

URL query:	https://rinkeby.etherscan.io/tx/
Smart contract	
transactionHash:	
”0xc8867b49c11bf52d4a065c5daefadb9f79ac6866fbb48feab5ef252abdaa0859”	
txIndex:	7
blockNumber:	4183700
gasUsed:	510710
hashIDJudge()	
transactionHash:	
”0x4727abeb77fda42a2d0dd31c6e9fceef5b89b06dab3e251d893f54c8293484cf”	
txIndex:	8
blockNumber:	4183728
gasUsed:	32650
lshvJudge()	
transactionHash:	
”0xca6b55091747a04270e985e8115bb7b4207dbd4d6e518d3ed3481d86e803fad4”	
txIndex:	3
blockNumber:	4183734
gasUsed:	33363



(a) Local Ganache

(b) Rinkeby Testnet

Figure 5: Execution time on local Ganache and Rinkeby testnet

they quickly receive financial rewards.

6. CONCLUSION

In this paper, the copyright detection architecture is based on blockchain, SHA256, LSH, and IPFS. The blockchain is used to store the hash values of legal copyrights and the IPFS is used to store the complete media. The SHA256 and the LSH are used in the detection stage. By implementing the proposed DCDChain, the media providers can obtain a credible detection method and the detection agent can also obtain a reasonable service fee. In the future, digital copyright detection technology will provide stronger support with the improvement of the hash algorithm.

References

References

[1] E. Sanchez, C. Russell, Sci-hub unmasked: Piracy, information policy, and your library, College and Research Libraries News.

[2] D. Gupta, et al., Study on extrinsic text plagiarism detection techniques and tools, Journal of Engineering Science & Technology Review 9 (5).

[3] K. Vani, D. Gupta, Integrating syntax-semantic-based text analysis with structural and citation information for scientific plagiarism detection, Journal of the Association for Information Science and Technology 69 (11) (2018) 1330–1345.

[4] A. Ekbal, S. Saha, G. Choudhary, Plagiarism detection in text using vector space model, in: 2012 12th International Conference on Hybrid Intelligent Systems (HIS), IEEE, 2012, pp. 366–371. doi:10.1109/HIS.2012.6421362.

[5] E. Y. Chang, J. Z. Wang, C. Li, G. Wiederhold, Rime: A replicated image detector for the world wide web, in: Multimedia Storage and Archiving Systems III, Vol. 3527, International Society for Optics and Photonics, 1998, pp. 58–68.

[6] Z. Zhou, C.-N. Yang, B. Chen, X. Sun, Q. Liu, J. QM, Effective and efficient image copy detection with resistance to arbitrary rotation, IEICE Transactions on information and systems 99 (6) (2016) 1531–1540. doi:10.1587/transinf.2015EDP7341.

[7] A. Wary, A. Neelima, A review on robust video copy detection, International Journal of Multimedia Information Retrieval (2018) 1–18doi:10.1007/s13735-018-0159-x.

[8] S. Sharma, C. S. Sharma, V. Tyagi, Plagiarism detection tool "parikshak", in: 2015 International Conference on Communication, Information & Computing Technology (ICCICT), IEEE, 2015, pp. 1–7.

[9] D. Weber-Wulff, C. Möller, J. Touras, E. Zincke, Plagiarism detection software test 2013, Abgerufen am 12 (2013) 2014.

[10] N. Nizamuddin, H. Hasan, K. Salah, R. Iqbal, Blockchain-based framework for protecting author royalty of digital assets, Arabian Journal for Science and Engineering (2019) 1–18doi:10.1007/s13369-018-03715-4.

[11] J.-C. Cheng, N.-Y. Lee, C. Chi, Y.-H. Chen, Blockchain and smart contract for digital certificate, in: 2018 IEEE international conference on applied system invention (ICASI), IEEE, 2018, pp. 1046–1051. `doi:10.1109/ICASI.2018.8394455`.

[12] A. Savelyev, Copyright in the blockchain era: promises and challenges, Computer law & security review 34 (3) (2018) 550–561. `doi:10.1016/j.clsr.2017.11.008`.

[13] S. Fujimura, H. Watanabe, A. Nakadaira, T. Yamada, A. Akutsu, J. J. Kishigami, Bright: A concept for a decentralized rights management system based on blockchain, in: 2015 IEEE 5th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), IEEE, 2015, pp. 345–346. `doi:10.1109/ICCE-Berlin.2015.7391275`.

[14] R. Xu, L. Zhang, H. Zhao, Y. Peng, Design of network media's digital rights management scheme based on blockchain technology, in: 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), IEEE, 2017, pp. 128–133.

[15] D. Bhowmik, T. Feng, The multimedia blockchain: A distributed and tamper-proof media transaction framework, in: 2017 22nd International Conference on Digital Signal Processing (DSP), IEEE, 2017, pp. 1–5. `doi:10.1109/ICDSP.2017.8096051`.

[16] J. Smith, M. Trubestein, Transacting real estate title using blockchain technology.

[17] Z. Ma, M. Jiang, H. Gao, Z. Wang, Blockchain for digital rights management, Future Generation Computer Systems 89 (2018) 746–764. `doi:10.1016/j.future.2018.07.029`.

[18] Z. Meng, T. Morizumi, S. Miyata, H. Kinoshita, Design scheme of copyright management system based on digital watermarking and blockchain,

in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Vol. 2, IEEE, 2018, pp. 359–364.

[19] R. Mehta, N. Kapoor, S. Sourav, R. Shorey, Decentralised image sharing and copyright protection using blockchain and perceptual hashes, in: 2019 11th International Conference on Communication Systems & Networks (COMSNETS), IEEE, 2019, pp. 1–6.

[20] S. Nakamoto, et al., Bitcoin: A peer-to-peer electronic cash system.

[21] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey, International Journal of Web and Grid Services 14 (4) (2018) 352–375. `doi:10.1504/IJWGS.2018.095647`.

[22] M. Datar, N. Immorlica, P. Indyk, V. S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the twentieth annual symposium on Computational geometry, ACM, 2004, pp. 253–262. `doi:10.1145/997817.997857`.

[23] M. Zhaofeng, H. Weihua, G. Hongmin, A new blockchain-based trusted drm scheme for built-in content protection, EURASIP Journal on Image and Video Processing 2018 (1) (2018) 91. `doi:10.1186/s13640-018-0327-1`.