This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610

HADES: Hash-based Audio Copy Detection System for Copyright Protection in Decentralized Music Sharing

Muhammad Rasyid Redha Ansori, Allwinnaldo, Revin Naufal Alief,
Ikechi Saviour Igboanusi, Jae Min Lee *Member, IEEE*, Dong-Seong Kim *Senior Member, IEEE*

***Abstract*—Preventive measures to stop copyright infringement are yet to be implemented on current decentralized music-sharing platforms. There is no mechanism to reject modified audio before they go online, so some decentralized music platforms become places full of pirated audio files. To address this problem, a perceptual hash-based audio detection method for copyright protection in decentralized music sharing was proposed. Chromaprint, an open-source audio fingerprint program, generates a hash value to detect copyright infringement. To assess the perceptual hash technique's robustness, Chromaprint generates a hash value from an audio file that can be modified with several signal processing attacks. The results of the detection system show that Chromaprint is very effective at spotting copyright infringement, with an average match rate of 92.64%. Deployed using a public Ethereum blockchain test network, the execution time from hashing to uploading to the IPFS distributed storage is only 616.3 ms.**

***Index Terms*—Audio, Blockchain, Copyright Protection, IPFS, Perceptual Hash.**

I. Introduction

DUE to developments in internet technology, everyone can enjoy and access multimedia content such as text, video, images, or audio on data-sharing websites. In addition, multimedia content has grown in popularity over the past few years [1]. All of this depends on the efforts of content creators, who constantly create and upload new content to the Internet to keep viewers entertained. Nonetheless, despite all of the Internet's improvements, these content creators encounter challenges in monetizing their works. Copyright infringement is one of the most insidious challenges in this industry, particularly in the music industry, which suffers significant economic losses due to piracy [2]. Duplicating or making minor modifications to content and receiving credit for the work, while the original creator receives no credit, is one of the most common instances of copyright infringement. Consequently, it is critical to detect copyright infringement to protect artists' intellectual property rights from their work.

Blockchain technology has been applied in various industries besides finance, such as information and security, supply chain management, and healthcare [3]–[5]. Blockchain establishes a shared distributed ledger based on the consensus mechanism among all network participants. Using a third-party verifier is no longer necessary, resulting in a secure and completely decentralized system. Thus, the system is decentralized, safe, and has shared tamper-proof ledgers that cannot be modified [6]. Blockchain technology can address the implementation concerns of trust, efficiency, privacy, and data sharing owing to its resilient and decentralized infrastructure [7]. According to Li et al. [8], blockchain technology can improve copyright protection and support content creators, which is advantageous for the future development of the music industry.

Although copyright management has advanced, the current decentralized copyright protection is not flawless. As reported by Gruhier [9], Audius, one of the most prominent decentralized music platforms, allows users to submit audio files to their system. Audius asserts that no content on their platform can be censored or erased. Allowing users to upload anything can create significant problems for agencies and musicians if no copyright claims system exists. In addition, Deahl [10] stated that there is no mechanism on the website for filing an infringement claim. Owing to the lack of audio detection technology on their platform, recent article [11] claimed that Audius has a high level of piracy, with much illegal content on the site. Hence, scanning content before it goes online is critical for keeping copyright-infringing content off existing decentralized music-sharing platforms.

The perceptual hash function is a method to identify multimedia data content. In contrast to the conventional hash function, which is highly sensitive to changes, the perceptual hash function is designed not to change significantly when multimedia data undergo minor modifications, so a slight change or manipulation in the input produces a slightly different output. In other words, the perceptual hash function can be treated as a unique identifier for each audio. Perceptual hashes have also proven their effectiveness in detecting modified multimedia content [12], with one of the most popular applications implementing perceptual hashing being Shazam.

This study proposes a system to detect manipulated audio for copyright protection in a decentralized music sharing platform. A perceptual hash is used to measure the similarity of audio with the existing audio in the system. If the audio file has similarities beyond the threshold, the system rejects the audio and lets the user know that the audio already exists.

Manuscript accepted on January 11, 2023. M. R. R. Ansori, Allwinnaldo, R. N. Alief, I. S. Igboanusi, J. M. Lee and D. -S. Kim are with the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Republic of Korea. Email: {rasyidred, winnaldo, revinnaufal, ikechisaviour, ljmpaul, dskim}@kumoh.ac.kr.

This paper was supported by Priority Research Centers Program (2018R1A6A1A03024003) and Basic Research Project (NRF-2022R1I1A3071844) of NRF, and the Grand Information Technology Research Center Support Program (IITP-2023-2020-0-01612) of IITP.

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610

TABLE I
LIST OF ACRONYMS USED IN THIS STUDY

Acronym	Description
AES	Advanced Encryption Standard
CNN	Convolutional Neural Network
DAG	Directed Acyclic Graph
DRM	Digital Rights Management
IPFS	InterPlanetary File System
LPF	Low-pass Filter
SNR	Signal to Noise Ratio
STFT	Short-time Fourier Transform
SVD	Singular Value Decomposition
URI	Uniform Resource Identifier

If the audio file similarity is less than the threshold, the audio is saved to the IPFS as off-chain storage. After acquiring an IPFS URI, they are saved in the Ethereum blockchain network through a smart contract. We also measure the robustness of the audio perceptual hash, audio detection, and uploading time.

The primary contributions of this paper are as follows:

1) We propose a model for detecting manipulated audio in a decentralized file storage system (IPFS) using a perceptual hash.
2) The proposed model can ensure audio availability, immutability, transparency, and protection of copyright protection in a decentralized music platform.
3) To the best of our knowledge, we are the first to implement blockchain-based audio file detection using a perceptual hash.

The remainder of this paper is organized as follows. Section II presents related work. Section III describes our proposed method. Section IV provides the results and Section V concludes the paper. Common acronyms used in this paper is listed on Table I.

II. RELATED WORK

This section discusses concepts and previous studies discussing the use of blockchain technology, particularly for detecting copyright infringement within multimedia data. Table II summarizes recent publications.

A. Machine Learning Approach

Several machine-learning approaches have been proposed to detect audio manipulation. For example, Lin et al. [13] introduced an audio tampering detection framework that uses supervised learning. Their framework uses statistical autoregressive features to identify whether the audio has been tampered with. The model achieved high accuracy against high noise addition and MP3 Compression. Das et al. [19] proposed a model that utilizes long-range acoustic features to detect spoofing attacks from audio speech data; their approach uses a deep neural network to separate genuine and spoofed speech data. Chenyu et al. [17] implemented a transformer-based CNN to detect audio spoofing, with a better result than that of the previous study. In paper [18], Wang et al. presented a framework that uses deep learning to detect tampered audio. Deep learning methods learn shallow and deep features from the audio file, which can improve detection accuracy. However, no studies have used blockchain technology to detect altered audio and cover a wide range of audio signal-processing attacks.

Furthermore, recent machine learning approaches for audio manipulation detection are unsuitable for decentralized file storage such as IPFS. IPFS breaks the file into small parts of data, hashes these data independently, forms Merkle DAGs, and then distributes them to peer nodes, making it very challenging to detect manipulated audio.

B. Watermarking Approach

In study [14], Zhao and O'Mahony advocated the use of Ethereum blockchain and smart contracts to protect music copyright using BMCProtector. There are three ways BMCProtector protects the music file: AES encryption, a watermarking method called vector quantization, and a DRM system to protect the music copyright off-chain. The smart contract oversees the exchange of the copyright information of music owners and the distribution of licensing fees to the wallet accounts of various copyright owners. However, the authors did not demonstrate the accuracy of their watermarking techniques in detecting audio files. In addition, because there is always a trade-off between robustness, imperceptibility, and capacity in watermarking, the choice of watermarking methods can be difficult, especially for audio files.

C. Fingerprint and Perceptual Hash Approach

Fingerprinting is a method that uniquely represents data characteristics and can be used to identify data [20]. A perceptual hash is a part of fingerprinting technology, where the term ”perceptual” means that the produced hash value should be consistent with what humans see or hear. The characteristic of perceptual hash is the produced hash does not change significantly where there is a change in data. Perceptual hashes are usually used to identify the original data and are widely applied for content authentication, forgery detection, and similarity detection [21]. The following are recent fingerprint and perceptual hash applications utilized with blockchain technology to detect audio manipulation.

Agye et al. [15] proposed integrating the Hyperledger Fabric blockchain and IPFS to preserve digital media copyright. They also employed Fourier transform for audio detection and perceptual hashing for image and video detection. However, the study did not describe how the Fourier transforms and perceptual hash were designed for detecting manipulated multimedia data.

In paper [16], Juan et al. introduced RobustCPS, an audio copyright protection system based on blockchain technology. RobustCPS divides audio into blocks and generates fingerprints with SVD for each. The similarity detection step is performed using a smart contract that determines whether there are fingerprints with identical characteristics on the Ethereum blockchain network. The fingerprint is listed on the blockchain if no identical fingerprint is found. The proposed system is resistant to common signal-processing attacks. Nevertheless, this system does not consider using decentralized storage, such as IPFS, which would increase security and privacy threats.

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610

TABLE II
COMPARISON OF PROPOSED SYSTEM WITH EXISTING BLOCKCHAIN-BASED AUDIO COPYRIGHT DETECTION SYSTEM

Study	Year	Data	Detection Technique	Blockchain Technology	Decentralized Storage	Accuracy Measurement
[13]	2017	Audio	Supervised Learning	No	No	Yes
[14]	2018	Audio	Watemarking	Ethereum	Yes	No
[15]	2019	Audio, Image, Video	Fourier Transform (Audio), Perceptual Hash (Image, Video)	Consortium	Yes	No
[16]	2020	Audio	SVD-based Fingerprint	Ethereum	No	Yes
[17]	2021	Audio	CNN	No	No	Yes
[18]	2022	Audio	Deep Neural Network	No	No	Yes
HADES	2022	Audio	Perceptual Hash	Ethereum	Yes	Yes

D. *Cyberattacks on Audio Files*

Many websites and software packages contain services that can modify audio files, including their metadata. These services usually include various common and de-synchronization modifications to modify audio signals. Although these services make it very easy for musicians to produce what kind of audio they want, it is also possible for them to lose money due to piracy because someone else can also download and modify their audio with these services without prior permission from the original creator.

Hence, to prove that an algorithm is resilient to modifications, the robustness of the algorithm must be checked to determine whether the altered audio has a similar output to the original audio. A reliable robustness test should include a comprehensive list of potential attacks [22]. Therefore, this paper implements various common audio signal modification and de-synchronization attacks to modify the audio signals. Common audio signal modifications include LPF, noise addition, and MP3 and AAC compression; examples of de-synchronization attacks include time-scale modification and pitch-shifting. Fig. 1 and 2 illustrate how the modifications affect the audio in the time and spectrum domain. More details about the audio manipulation attacks are provided in Section IV-C.

The following explanations provide the details about each attack on audio files performed in this paper.

1) LPF: a filter that passes audio signals with a frequency lower than a selected cut-off frequency and attenuates signals with frequencies higher than the cut-off frequency.
2) Noise addition: noise is added to the audio file until the desired SNR is achieved.
3) Time scale modification: the tempo and audio duration is changed while maintaining pitch.
4) Pitch-shifting: the audio pitch is reduced while maintaining its tempo and duration.
5) MP3 compression: an MP3 encoder is used to lower the bitrate of the audio.
6) AAC compression: an AAC encoder is used to lower the bitrate of the audio.

III. PROPOSED SYSTEM

This section describes the details of the perceptual hash method (III-A) and detection of copyright infringement by the proposed system (III-B). The general scheme used in this system is illustrated in Fig. 3. The steps are as follows.

1) A node tries to upload an audio file into the system. The node user can be a musician or composer who wants to share their original work or a person who wants to steal audio and pass it off as their own.
2) The perceptual hash of the audio file is computed as a unique identifier. Chromaprint is used as the perceptual hash in this paper.
3) The acquired perceptual hash is compared to the existing hashes recorded in the blockchain network.
4) If the similarity of the hash values is less than 50%, the audio file is added to IPFS. IPFS returns a URI after adding the audio, and this URI is recorded in the blockchain network using a smart contract.
5) If the similarity of the hash values is more than 50%, a copyright infringement is detected, the upload process audio is rejected, and the blockchain network is not updated.

The criterion for the similarity of perceptual hash values is set at 50%, as some studies have utilized this threshold for their evaluations [23], [24].

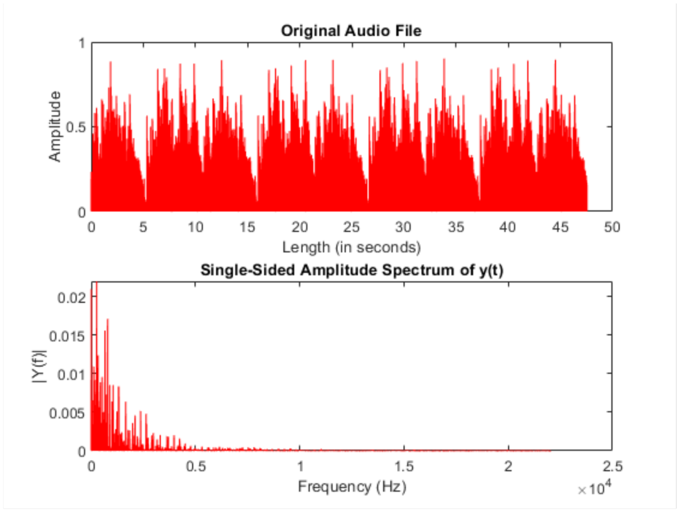
A. *Perceptual Hash Method*

This study's perceptual hash approach is based on Chromaprint, an open-source audio fingerprinting tool. Lukas Lalinsky designed Chromaprint as a library on the client side that extracts fingerprints from any audio source. The following is a step-by-step explanation of how Chromaprint generates perceptual hashes from audio [25], [26].

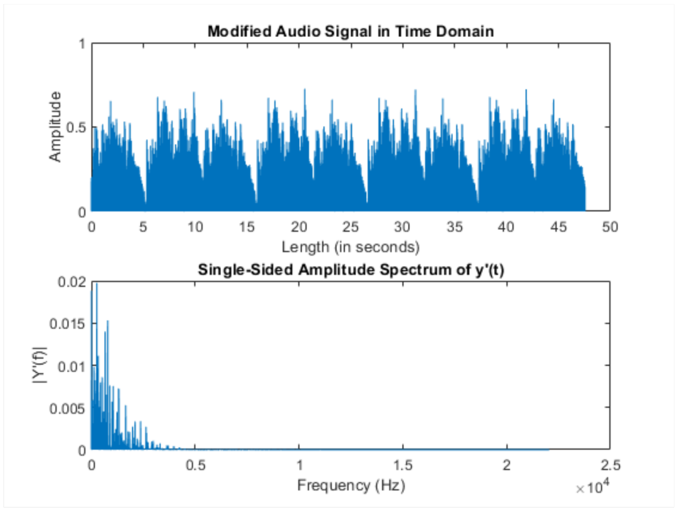
1) The input audio is converted to mono and downsampled to 11,025 Hz
2) The audio signal is transformed into the frequency domain using a STFT with a frame size of 4096 samples and a two-third overlap.
3) The spectrum is then divided into 12 bins. Each bin represents a different chroma value.
4) Six filters are generated to calculate the Chromaprint hash values.
5) Another 16 filters are generated using the AdaBoost method described by [27] with different sizes from the six filters.

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

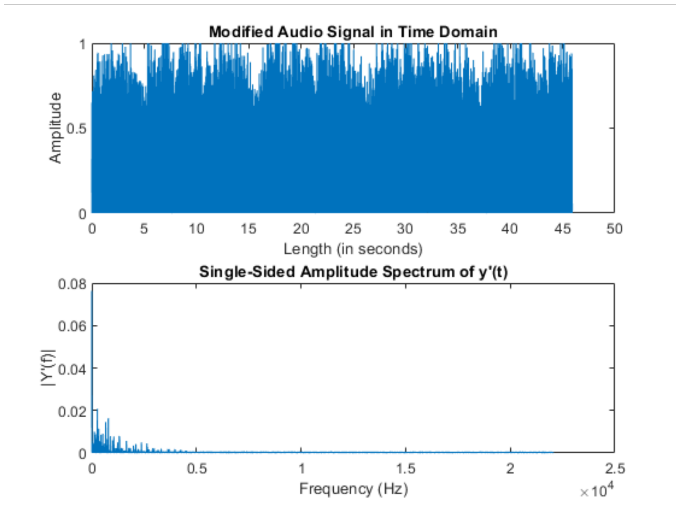
This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610



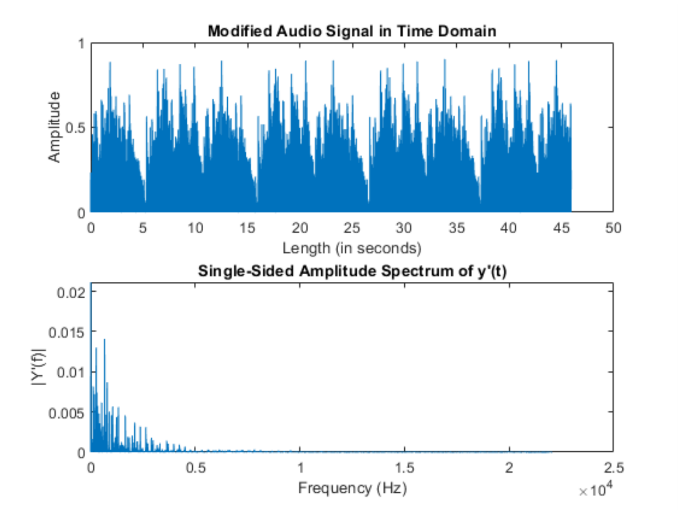
Original Audio



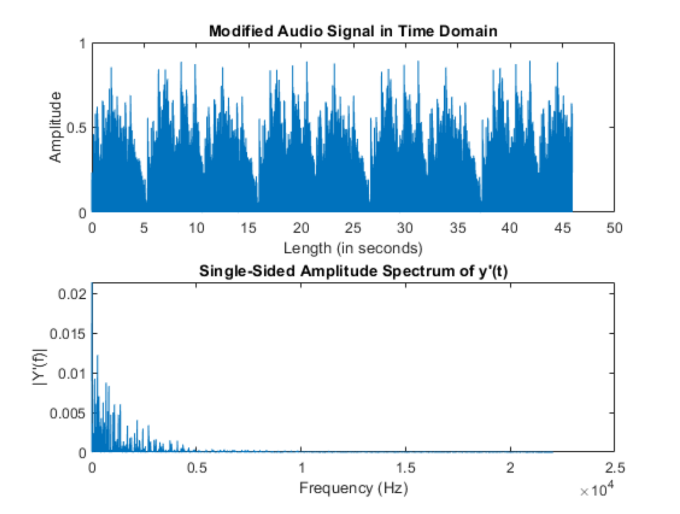
Modified with LPF 3KHz



Modified with Adding Noise 0dB



Modified with Time Scale Modification 0.96 times

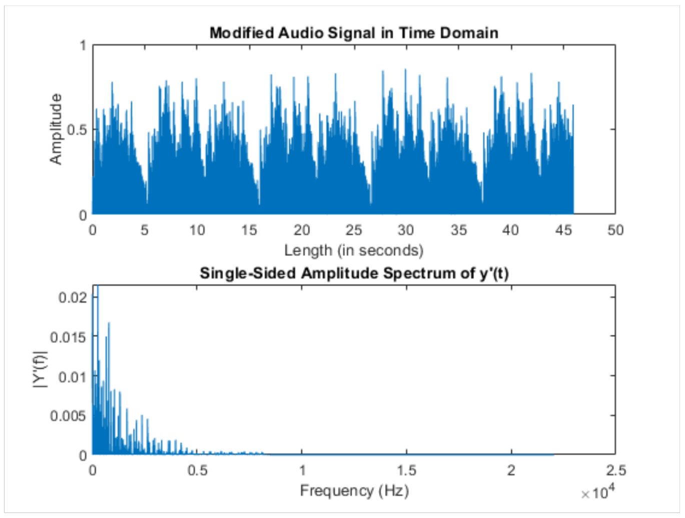


Modified with Pitch Shifting 0.96 times

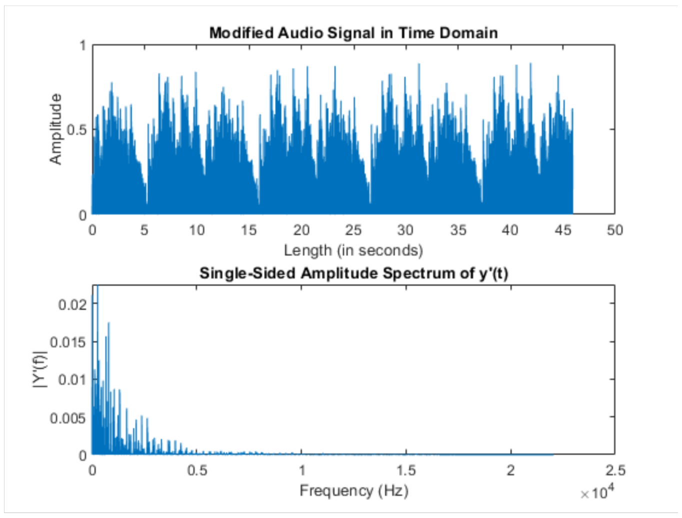
Fig. 1. Illustration of Original and Modified Audio: Low-pass Filter (LPF) of 3KHz, Noise of 0 dB, Time Scale Modification of 0.96 Times, and Pitch-Shifting of 0.96 Times.

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610



Modified with MP3 Compression 32KHz



Modified with AAC Compression 32KHz

Fig. 2. Illustration Modified Audio: MP3 Compression with 32KHz, AAC Compression with 32KHz

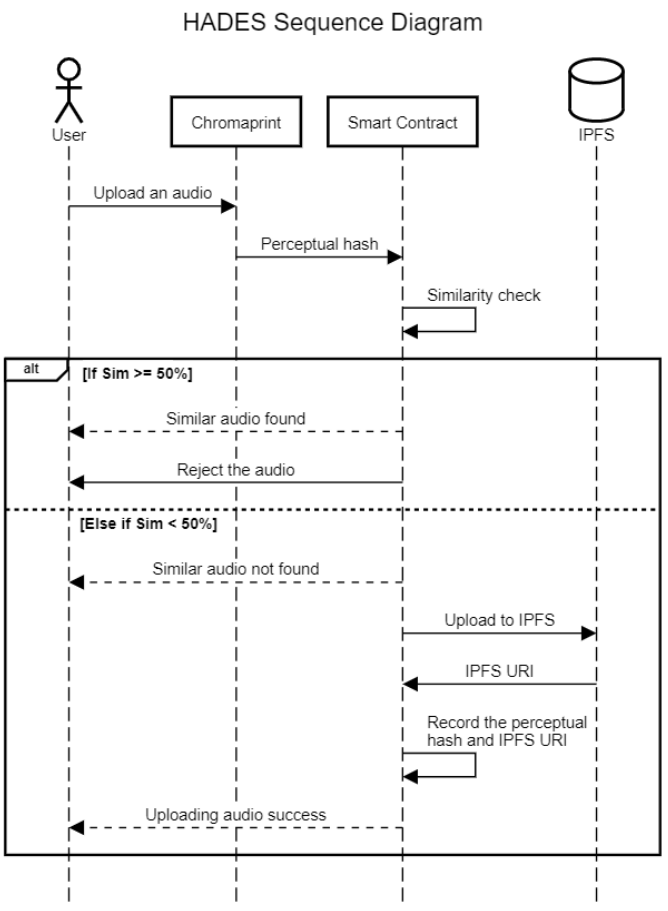


Fig. 3. Scheme of The Proposed System

6) A 12x16 sliding window slides across the chroma value on each sample, producing a sub-image.
7) For each frame, the window is filtered by the 16 generated filters.
8) A filter adds up the energy value in the white region and subtracts it from the black area, yielding a single value. Also, the energy value is encoded as a two-bit number (ranging from 0 to 3) using Gray code.
9) Repeat steps 7 and 8 for every sub-image, and an audio fingerprint is acquired.

B. Detection of Copyright Infringement on Audio Files

To detect audio similar to the existing ones in the system, the perceptual hash value of the audio must be generated first using the Chromaprint method. The pseudocode of generating perceptual hash from audio is illustrated in Algorithm 1.

Algorithm 1: Generating Perceptual Hash of an Audio using Chromaprint

```
Result: Hash Value of An Audio
initialization;
import chromaprint;
import pyacoustid;
audioPHash ← 0;
audioPHash ← getFingerprint(audiofile);
Return audioPHash;
(continue to algorithm 2)
```

Algorithm 1 shows how a perceptual hash can be acquired from the audio. First, select the desired audio file to be added to the system is entered. The $getFingerprint$ *function* from Chromaprint generates a perceptual hash of the selected audio that stores the value into the $audioPHash$ variable. After acquiring the hash value, the $audioPHash$ value is compared with the existing perceptual hash value from the blockchain network using the Hamming distance rule in algorithm 2.

The purpose of Algorithm 2 is to calculate the similarity between the perceptual hash of the candidate audio file and the perceptual hashes stored in the blockchain network. Before using the Hamming distance method, the $audioPHash$ value is stored into $h1$, all of the stored perceptual hashes from the blockchain network are extracted ($hades.phashList$) and saved into variable $audioPHashList$, an empty array $percentage$ is created to save the similarity value of $h1$ and

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610

Algorithm 2: Audio Detection

```
Result: Audio Acceptance or Rejection
initialization;
h1 ← audioPHash;
audioPHashList ← hades.phashList;
percentage ← [];
similar ← FALSE;
i ← 1;
//Hamming Distance //
while $i \leq$ length(audioList) do
    h2 ← audioPHashList[i];
    minLen ← min(length(h1),length(h2));
    similar ← FALSE;
    distance ← 0;
    k ← 1;
    while $k \leq$ minLen do
        if $h1[k] \neq h2[k]$ then
            distance ← distance + 1;
        else
            continue;
        end
        k ← k + 1
    end
    i ← i + 1;
    percentage[i] ← (distance×100)÷minLen;
end
j ← 1;
while $j \leq$ length(percentage) do
    if percentage[j] $\geq 50\%$ then
        similar ← TRUE;
        reject the audio;
    else
        continue;
    end
    j ← j + 1;
end
upload to IPFS (continue to algorithm 3);
```

each of $audioPHashList$, and the Boolean value of the variable $similar$ that indicates whether the candidate audio is similar to the audio files in the system is set to FALSE. Then, check the similarity between $h1$ and $audioPHashList$ one by one. In the first loop, the first value of $audioPHashList$ is saved in $h2$ and the minimum lengths ($minLen$) of $h1$ and $h2$ are calculated. The Hamming distance is then used to obtain the similarity percentage by matching the perceptual hash values bit by bit until they reach $minLen$, which indicates the bit places where the two values have the same value. Suppose the similarity between the candidate audio and the audio files in the system is equal to or greater than 50%. In that case, the system rejects adding the audio to the system. However, if the similarity is less than 50%, looping continues until the end of the audio list in $audioPHashList$.

After the similarity check process is completed, using algorithm 3, the audio file and the audio's perceptual hash are stored in the IPFS, which then returns a URI ($Iaudio$). Finally, $Iaudio$ is saved to the Ethereum blockchain network using a smart contract, along with $h1$.

Algorithm 3: Save Audio To IPFS & Its URI And Hash To Smart Contract

```
//Connecting to the IPFS Storage//;
ipfs.start();
//Adding Audio File to IPFS and Its URI to Smart
  Contract//;
hades.Iaudio ← ipfs.add(audiofile);
hades.PHashList ← [hades.PHashList, h1];
//Close the IPFS connection//;
ipfs.stop();
```

IV. Experimental Setup and Result

This section discusses the details of the experimental setup (IV-A), execution time (IV-B), and the robustness of the perceptual hash technique against signal processing attacks (IV-C).

A. Experimental Setup

The simulation in this paper was carried out on a Macbook Air powered by the M1 chip with an 8-core GPU, 8 GB RAM, and 512 GB storage. The audio file tested for the proposed approach had a 16-bit resolution, 47 s duration, and a 44, 100 Hz sample rate. Audio manipulation was performed using MATLAB 2021a. The blockchain was configured using Ethereum blockchain and tested in the Goerli public test network. Finally, the smart contract was written in Solidity.

B. Perceptual Hash Execution Time

In this section, we discuss the execution time of the proposed system. The proposed system was tested by measuring the time required for the perceptual hashing process, audio detection, upload of the file to the IPFS, and updating the data to the Ethereum blockchain network. Table III shows the average execution time of the proposed system, and Table IV shows the comparison of the hashing average times between the proposed system and previous studies.

Based on Table III, the hashing time using the Chromaprint program was approximately 156 ms. The detection time to

TABLE III
Average HADES Audio Detection Time (ms)

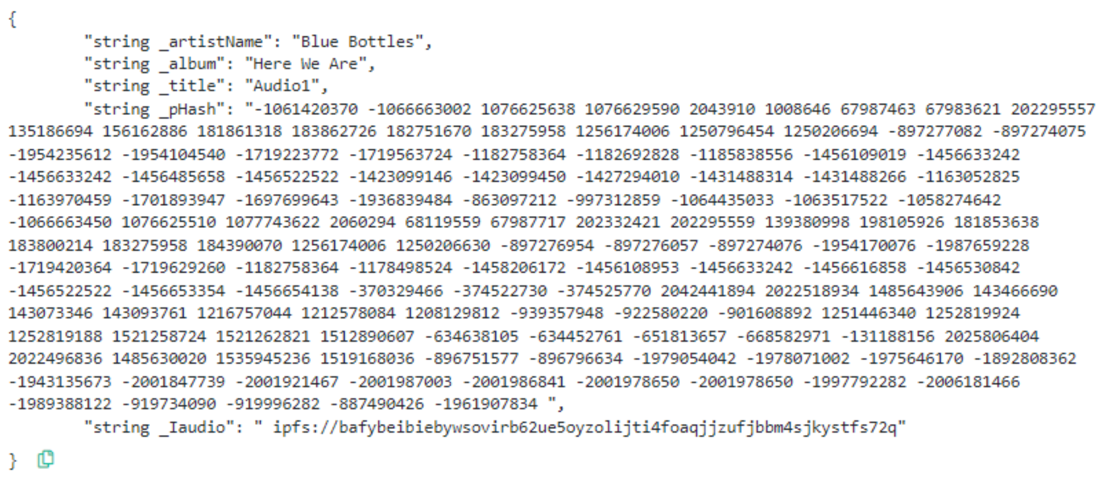
Hash Function	Similarity Check	Upload to IPFS	Total Time
156	168.3	292	616.3

TABLE IV
Method Comparison of Hashing Average Time (ms)

Method	Average Time
RobustCPS [16]	2,776
HADES	156

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610



```
{
        "string _artistName": "Blue Bottles",
        "string _album": "Here We Are",
        "string _title": "Audio1",
        "string _pHash": "-1061420370 -1066663002 1076625638 1076629590 2043910 1008646 67987463 67983621 202295557
135186694 156162886 181861318 183862726 182751670 183275958 1256174006 1250796454 1250206694 -897277082 -897274075
-1954235612 -1954104540 -1719223772 -1719563724 -1182758364 -1182692828 -1185838556 -1456109019 -1456633242
-1456633242 -1456485658 -1456522522 -1423099146 -1423099450 -1427294010 -1431488314 -1431488266 -1163052825
-1163970459 -1701893947 -1697699643 -1936839484 -863097212 -997312859 -1064435033 -1063517522 -1058274642
-1066663450 1076625510 1077743622 2060294 68119559 67987717 202332421 202295559 139380998 198105926 181853638
183800214 183275958 184390070 1256174006 1250206630 -897276954 -897276057 -897274076 -1954170076 -1987659228
-1719420364 -1719629260 -1182758364 -1178498524 -1458206172 -1456108953 -1456633242 -1456616858 -1456530842
-1456522522 -1456653354 -1456654138 -370329466 -374522730 -374525770 2042441894 2022518934 1485643906 143466690
143073346 143093761 1216757044 1212578084 1208129812 -939357948 -922580220 -901608892 1251446340 1252819924
1252819188 1521258724 1521262821 1512890607 -634638105 -634452761 -651813657 -668582971 -131188156 2025806404
2022496836 1485630020 1535945236 1519168036 -896751577 -896796634 -1979054042 -1978071002 -1975646170 -1892808362
-1943135673 -2001847739 -2001921467 -2001987003 -2001986841 -2001978650 -2001978650 -1997792282 -2006181466
-1989388122 -919734090 -919996282 -887490426 -1961907834 ",
        "string _Iaudio": " ipfs://bafybeibiebywsovirb62ue5oyzolijti4foaqjjzufjbbm4sjkystfs72q"
}
```

Fig. 4. Transaction Illustration in The Smart Contract

confirm whether the input audio was similar to the existing ones in the system was approximately 168.3 ms. The average time required to upload an audio file and its perceptual hash value was 292 ms in total. However, the uploading time depends on the user's internet bandwidth, so it may take longer to upload files if there is an internet connection problem. Hence, the total execution time for the audio detection is 616.3 ms. Furthermore, Table IV shows HADES demonstrates a much faster hashing process than RobustCPS, which has an average hashing time of 2,776 ms. However, this quick hashing process also comes with a drawback; less accurate detection, as discussed in the following section.

C. Detection of Modified Audio Files

Various signal-processing attacks are applied to audio files to evaluate the robustness of the Chromaprint perceptual hash in detecting copyright infringement. The parameters for each manipulation are set as follows.

1) LPF: LPF was implemented using the 2^{nd} order Butterworth filter with cut-off frequencies of 3, 6, and 9 kHz.
2) Noise addition: the amount of noise was increased until the SNR reached 0, 10, and 20 dB.
3) Time scale modification: the tempo was slowed with a modifier of 99%, 98%, 97%, and 96%.
4) Pitch-shifting: the pitch was decreased to 99%, 98%, 97%, and 96% from its normal level.
5) MP3 and AAC compression: the audio bitrate was compressed to 32, 64, and 128 Kbps.

After applying the signal processing attacks to the original audio, the perceptual hashes of the original and attacked audio were compared using the Hamming distance (see algorithm 2). Higher similarity indicates better performance in identifying modified audio. Table V shows the accuracy of the proposed system and the system cited in [16] (RobustPCS) for detecting copyright infringement after applying signal-processing attacks to the original audio. The performance of RobustCPS is slightly better, with an average similarity of 3.05% higher than HADES.

There is little difference in performance when detecting audio filtered by LPF and compressed by MP3 and AAC encoders. However, a performance gap is evident regarding noise addition and pitch-shifting manipulation. Adding noise until the SNR reaches 0 dB has a significant effect on the audio signal, with the most noticeable difference occurring in the time domain (as shown in Fig. 1), making it difficult for both RobustCPS and HADES to detect the audio. However, HADES has considerably better results than RobustCPS in detecting noise-added audio, with similarities of 83.95% and 76.34%, respectively. In contrast, HADES is inferior to RobustCPS in pitch-shifting manipulation, especially when the audio's pitch is changed to 96%. HADES can only achieve a similarity level of 77.93%, in contrast to RobustCPS's ability which is able to obtain a similarity value of 99.05%. The pitch-shifting manipulation affected the frequency of the audio stream (as shown in Fig. 1), making it difficult for HADES to identify the altered audio perfectly. Therefore, HADES performs better in detecting noise-added audio but worse in detecting pitch-shifted audio than RobustCPS.

D. HADES Smart Contract

As described in the experimental setup (section IV-A), the smart contract for the proposed system was written in Solidity and deployed to the Goerli public test network. Table VI shows the transaction fee of the HADES smart contract for each function, and Fig. 4 illustrates the content of transaction in the smart contract captured from Remix IDE.

Table VI shows the highest costs when executing $registerAudio()$, a function that registers an audio file after successfully passing the similarity check process, with a fee of 3,183,947 Gas. Compared to $registerAudio()$, contract deployment has only a transaction fee of 531,160 Gas. The reason why $registerAudio()$ is costlier than deploying the smart contract to the blockchain network is that storing strings in the blockchain is more expensive than storing other types of variables. $getURI()$ is a function for retrieving the audio's IFPS URI, whereas $getpHash()$ returns the audio's perceptual hash value that is stored in the blockchain. $getAllInfo()$ is

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610

TABLE V
METHOD COMPARISON OF PERCEPTUAL HASH FUNCTION TO DETECT MANIPULATED AUDIO

Attacks	Parameter	Similarity (%)	
		RobustCPS	HADES
LPF	3 kHz	96.84	96.94
	6 kHz	99.05	99.66
	9 kHz	99.37	99.97
Noise Addition	0 dB	76.34	83.95
	10 dB	96.21	97.22
	20 dB	99.68	99.44
Time Scale Modification	99%	99.05	98.40
	98%	98.74	97.29
	97%	98.74	97.04
	96%	99.68	96.51
Pitch-Shifting	99%	99.37	83.26
	98%	99.68	78.96
	97%	99.05	78.31
	96%	99.05	77.93
MP3 Compression	32 kbps	90.20	92.55
	64 kbps	90.85	92.75
	128 kbps	90.85	92.67
AAC Compression	32 kbps	93.67	96.61
	64 kbps	93.67	96.61
	128 kbps	93.67	96.61
Average Similarity		95.69	92.64

TABLE VI
TRANSACTION FEE OF HADES SMART CONTRACT

Functions	Transaction Fee (Gas)	Transaction Fee (ETH)
Deploying Contract	531,160	0.01594
$registerAudio()$	3,183,947	0.09552
$getURI()$	0	0
$getpHash()$	0	0
$getAllInfo()$	0	0

used to return all information regarding the audio, including perceptual hash value, and IPFS URI. The last three functions do not require fees to invoke them, as they are only getter functions.

V. CONCLUSION AND FUTURE WORK

This study presented a robust audio detection system for copyright protection in decentralized music sharing. In the proposed system, a perceptual hash is used to detect copyright infringement in audio files. The perceptual hash technique used in this research was Chromaprint, an open-source audio fingerprint library developed for the AcoustID project. The proposed system was built on the Ethereum blockchain and IPFS to securely record musicians and their music and keep audio files available thanks to the blockchain's security, transparency, immutability, and IFPS advantage. The results demonstrated that the proposed system is highly resistant to signal processing attacks, such as LPF, noise addition, time scaling modification, pitch-shifting, MP3 compression, and AAC compression. In this paper, the threshold value for the similarity was set to 50%. The average similarity between the original audio and its modified versions was not less than 90% for any scenario, which means that the proposed system could effectively detect copyright infringement. Furthermore, the average audio detection time of the proposed system was only 616.3 ms. For further research, it is necessary to improve the perceptual hashing technique to cover a broader range of attack resistances.

REFERENCES

[1] M.-J. Kim, C. Yoo, and Y.-W. Ko, "Multimedia File Forensics System Exploiting File Similarity Search," *Multimedia Tools and Applications*, vol. 78, pp. 5233–5254, 03 2019.

[2] P. Rai, "Copyright Laws and Digital Piracy in Music Industries: The Relevance of Traditional Copyright Laws in the Digital Age and How Music Industries Should Cope with The Ongoing Piracy Culture," Ph.D. dissertation, Universitetet i Agder, 02 2021.

[3] I. S. Igboanusi, K. P. Dirgantoro, J.-M. Lee, and D.-S. Kim, "Blockchain Side Implementation of Pure Wallet (PW): An Offline Transaction Architecture," *ICT Express*, vol. 7, no. 3, pp. 327–334, 2021.

[4] K. P. Dirgantoro, J. M. Lee, and D.-S. Kim, "Generative Adversarial Networks Based on Edge Computing With Blockchain Architecture for Security System," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 039–042.

[5] N. Nizamuddin, K. Salah, M. Ajmal Azad, J. Arshad, and M. Rehman, "Decentralized Document Version Control using Ethereum Blockchain and IPFS," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[7] J. Zhao, S. Fan, and J. Yan, "Overview of business innovations and Research Opportunities in Blockchain and Introduction to The Special Issue," *Financial Innovation*, vol. 2, 12 2016.

[8] Y. Li, J. Wei, J. Yuan, Q. Xu, and C. He, "A Decentralized Music Copyright Operation Management System Based On Blockchain Technology," *Procedia Computer Science*, vol. 187, pp. 458–463, 2021, 2020 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI2020.

[9] C. Gruhier, "Audius: The Streaming Platform That Pays Its Artists (and Listeners) The Most?" Mar 2021. [Online]. Available: https://www.haumeamagazine.com/en/audius-the-streaming-platform-that-pays-its-artists-and-listeners-the-most/

[10] D. Deahl, "New Blockchain-Based Music Streaming Service Audius is A Copyright Nightmare," Oct 2019. [Online]. Available: https://www.theverge.com/2019/10/9/20905384/audius-blockchain-music-streaming-service-copyright-infringement-piracy

[11] A. Jones, "Audius' Next-generation Streaming Service is Plagued by Piracy: Billboard," Feb 2022. [Online]. Available: https://www.nmpa.org/audius-next-generation-streaming-service-is-plagued-by-piracy/

[12] P. Samanta and S. Jain, "Analysis of Perceptual Hashing Algorithms in Image Manipulation Detection," *Procedia Computer Science*, vol. 185, pp. 203–212, 2021.

[13] X. Lin and X. Kang, "Supervised audio tampering detection using an autoregressive model," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2142–2146.

[14] S. Zhao and D. O'Mahony, "BMCProtector: A Blockchain and Smart Contract Based Application for Music Copyright Protection," in *Proceedings of the 2018 International Conference on Blockchain Technology and Application*, ser. ICBTA 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–5.

[15] K. O.-B. O. Agyekum, Q. Xia, Y. Liu, H. Pu, C. N. A. Cobblah, G. A. Kusi, H. Yang, and J. Gao, "Digital Media Copyright and Content Protection Using IPFS and Blockchain," in *Image and Graphics*, Y. Zhao, N. Barnes, B. Chen, R. Westermann, X. Kong, and C. Lin, Eds. Cham: Springer International Publishing, 2019, pp. 266–277.

[16] J. Zhao, T. Zong, Y. Xiang, L. Gao, and G. Beliakov, "Robust blockchain-based cross-platform audio copyright protection system using content-based fingerprint," in *Web Information Systems Engineering – WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 201–212. [Online]. Available: https://doi.org/10.1007/978-3-030-62008-0_14

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3241610

[17] C. Liu, J. Li, J. Duan, H. Shen, and H. Huang, "LightCvT: Audio Forgery Detection via Fusion of Light CNN and Transformer," in *2021 10th International Conference on Computing and Pattern Recognition*, ser. ICCPR 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 99–105.

[18] Z. Wang, Y. Yang, C. Zeng, S. Kong, S. Feng, and N. Zhao, "Shallow and Deep Feature Fusion for Digital Audio Tampering Detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, pp. 1–20, 2022.

[19] R. K. Das, J. Yang, and H. Li, "Long range acoustic and deep features perspective on asvspoof 2019," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 1018–1025.

[20] Y. Jiang, C. Wu, K. Deng, and Y. Wu, "An Audio Fingerprinting Extraction Algorithm Based on Lifting Wavelet Packet and Improved Optimal-basis Selection," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30 011–30 025, 2019.

[21] X. Wang, X. Zhou, Q. Zhang, B. Xu, and J. Xue, "Image Alignment Based Perceptual Image Hash for Content Authentication," *Signal Processing: Image Communication*, vol. 80, p. 115642, 2020.

[22] Y. Lin, W. H. Abdulla *et al.*, *Audio watermark*. Springer, 2015, vol. 146.

[23] O. Uzuner, R. Davis, and B. Katz, "Using Empirical Methods for Evaluating Expression and Content Similarity," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, 2004, pp. 8 pp.–.

[24] R. Kumar, R. Tripathi, N. Marchang, G. Srivastava, T. R. Gadekallu, and N. N. Xiong, "A Secured Distributed Detection System based on IPFS and Blockchain for Industrial Image and Video Data Security," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 128–143, 2021.

[25] "Lukáš lalinský." [Online]. Available: https://oxygene.sk/2011/01/how-does-chromaprint-work/

[26] M. Chickanbanjar, "Comparative Analysis Between Audio Fingerprinting Algorithms," *International Journal of Computer Science & Engineering Technology (IJCSET)*, vol. 8, pp. 185 – 194, 05 2017.

[27] D. Jang, C. D. Yoo, S. Lee, S. Kim, and T. Kalker, "Pairwise Boosted Audio Fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 995–1004, 2009.

Muhammad Rasyid Redha Ansori earned his Bachelor's degree in Telecommunication Engineering from Telkom University, Indonesia in 2020. He is currently pursuing a Master's degree in IT Convergence Engineering and is a researcher at the Kumoh National Institute of Technology's Networked System Lab in Gumi, South Korea. His research interests include blockchain, smart contracts, information and security, real-time systems, and machine learning.

Revin Naufal Alief earned his Bachelor's degree in Telecommunication Engineering from Telkom University, Indonesia in 2020. He is pursuing a Master's degree in IT Convergence Engineering at the Kumoh National Institute of Technology, Gumi, South Korea. He is also a researcher in the Network System Laboratory, a laboratory in Kumoh National Institute of Technology. His research interests include blockchain, real-time systems, information and security, and machine learning.

Allwinnaldo earned his Bachelor of Engineering degree in Telecommunication Engineering from Telkom University, Indonesia in 2020. He is pursuing a master's degree in IT Convergence Engineering and as a researcher at Kumoh National Institute of Technology's Networked System Lab (NSL) in Gumi, South Korea. His research interests include blockchain, digital signal processing, audio watermarking, and machine learning.

Ikechi Saviour is currently working towards his PhD degree in IT Convergence Engineering, at Kumoh National Institute of Technology, Gumi, South Korea. He received his M.Eng degree from the department of IT Convergence Engineering at Kumoh National Institute of Technology, South Korea, in 2020. He received his B. Tech degree in Physics from the Federal University of Technology Owerri, Nigeria, in 2013. He has been a researcher in Networked Systems Lab since 2018 and has been an IEEE member since 2011. His interest research area includes blockchain, Network load balancing, real-time networks, and machine learning.

Jae-Min Lee received the Ph.D degree in electrical and computer engineering from the Seoul National University, Seoul, Korea, in 2005. From 2005 to 2014, he was a Senior Engineer with Samsung Electronics, Suwon, Korea. From 2015 to 2016, he was a Principle Engineer in Samsung Electronics, Suwon, Korea. Since 2017, he has been an assistant professor with School of Electronic Engineering and Department of IT-Convergence Engineering, Kumoh National Institute of Technology, Gyeongbuk, Korea. He is a member of IEEE. His current main research interests are industrial wireless control network, performance analysis of wireless networks, and TRIZ.

Dong-Seong Kim (SM'14) (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, Republic of Korea, in 2003. From 1994 to 2003, he worked as a full-time Researcher at the ERC-ACI, Seoul National University, Seoul. From March 2003 to February 2005, he worked as a Postdoctoral Researcher with the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently the Director with the ICT Convergence Research Center (Grand ICT Program), Kit Convergence Research Institute supported by the Korean Government with the Kumoh National Institute of Technology. He is a senior member of IEEE and ACM. His current main research interests are real-time IoT and smart platform, industrial wireless control network, networked embedded.

Authorized licensed use limited to: Universitas Indonesia. Downloaded on August 23,2023 at 00:24:15 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.