SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Atomic Structure Parsing: A Novel Framework for Digital Audio file Forgery Detection

Jashanjot Kaur[1,2], Abhinav Sood[3], Shivani Sharma[4]

[1] Department of Forensic Science, Chandigarh University, Gharuan, Mohali, Punjab, India-140413.
[2] Punjab Bureau of Investigation, Chandigarh, sector-9D Chandigarh,India-160009.
[3] Department of Forensic Science, Chandigarh University, Gharuan, Mohali, Punjab, India-140413.
[4] Central Forensic Science Laboratory, Chandigarh, DFSS, MHA, GoI-160036.
*Corresponding Author: E-mail: jashanjotkaur82@gmail.com

Abstract
Digital advancements, accessible free editing tools, and Artificial Intelligence have fundamentally transformed multimedia authentication challenges. Digital audio files being the primary medium of communication and crucial evidentiary material in the court trials, the authentication and the integrity check of the digital audio files is necessary to rule out any intentional forgery or tampering in them. The blind detection of the tampering in the absence of the recording media or reference audio file, is in itself an another major challenge which is being addressed through the method discussed in this paper. The paper presents an in-depth insight into the structure of the digital audio files, forgery detection and ultimately aims at determining the authenticity and thus integrity of the audio files based on the atomic structure parsing and analysis of the container of digital audio files and exploiting the metadata and the structural values of the container atoms thereof. The proposed method works efficiently to identify the tampered digital audio files altered by the means of cutting, copying, copying-pasting and deliberate noise addition modifications. By detecting post-manipulation re-indexing and re-muxing through container structure anomalies of addition, deletion or changes in the patterns in the atomic structure, the proposed approach empowers the forensic examiners to reliably differentiate authentic primary digital audio recordings from secondary or tampered derivatives for court proceedings.

Keywords: Forensic Science, Multimedia Forensics, Forensic Electronics, Audio Forensics, Forgery Detection, Authentication of Audio Files, Blind Detection Method, Parsing and Container Atom Analysis, Metadata Analysis, etc

Introduction

The rapid and easy proliferation of multimedia files, audio editing softwares and artificial intelligence technologies in the digital world has contributed to a significant number of crimes pertaining to the illicit possession, modification, and distribution of multimedia files especially digital audio files[1,2]. The authentication and the integrity check of such digital audio files is necessary to rule out any intentional tampering to prevent misinformation spreading and also as a pre-requisite for the audio recordings to be considered as an evidence in the court room[3]. In the vast and advanced ocean of technology, audio file editing or forgery can be achieved just by the use of very simple and handy software tools like adobe audition, audacity, studio One, Pro Tools, Cubase or using online authentic and unauthentic sources[4]. These are available in both as free and paid versions[5].

Essentially to verify the integrity of an audio file and hence to determine any kind of tampering thereof in the audio file, two type of approaches are used- Active detection technique and Passive detection technique (also called blind detection technique)[6]. Active detection technique basically rely on the extraction and analysis of some kind of hidden signature, watermark or code in the original recording that was embedded into the audio file at the time of its recording[7]. Authenticity and integrity is confirmed when the embedded and retrieved signature or watermark or code match; a discrepancy (alteration or elimination) indicates a the file was not original and it was either tampered with or underwent unintentional processiong[8]. But this technique requires that the original recording had some sort of such mark embedded at the time of its recording or saving and that this mark is known to the analyst, but in real life scenarios it is unknown. Passive detection technique on the other hand is like a detective looking at clues left behind on the basis of Locards Exchange Principle[9,10,11].The analyst analyse the audio file itself, without needing any prior information. Analyst analyses the inconsistencies, unusual patterns, or anything that doesn't seem natural. This is much more practical and also much harder because analyst tries to find very subtle signs of manipulation. Though Active detection technique is more reliable, but passive methods are more versatile for real-world scenarios where there are no clues about the origin or the source of the audio recording.

Spectrogram analysis was once the most reliable tool for audio authentication and is still being used. Spectrogram analysis allows analysts to easily detect tampering in audio files through the presence of discontinuities like- sudden breaks, drop outs or gaps in the spectrogram and Abrupt Changes in the frequency or amplitude patterns (indicating splicing or other forms of editing)[12,13]. Refer to Figure-1 where such an abruption is depicted via a red color line break observed in the spectrogram of the audio file segment. Besides these indicators, analysts also observe other indicators of tampering like inconsistencies in the background or environmental noise levels or patterns, presence of Unnatural Frequency Patterns due to manipulations like pitch shifting or time stretching and presence of sudden changes in the intensity of frequency bands that are very well indicative of the tampering in audio files through spectrogram analysis. Not only the spectrogram analysis reveals the status of forgery in audio files but it also indicated the precise location of the forged region in the audio file[14].

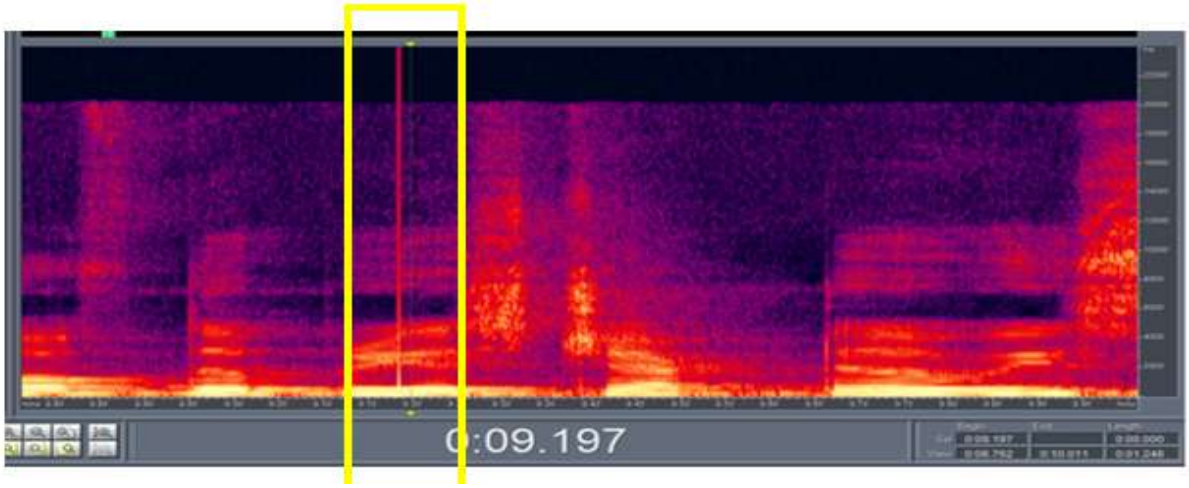

Figure 1: Spectrographic representation of an audio file segment highlighting a potential discontinuity (visible as red colored streak) indicative of a forged region in the digital audio file.

Spectrograms were once a powerful tool, but modern audio editing tools and techniques powered by sophisticated artificial intelligence have become so sophisticated that they can effectively "hide" the telltale signs of tampering [15,16]. Today advanced editing tools can perform incredibly precise audio manipulation. Editors can seamlessly blend audio segments, remove indicator artifacts, and alter frequencies in ways that leave minimal to no visual traces on a spectrogram for analyst to discover. Things like background noise, speaker characteristics, and environmental sounds can be replicated or altered with very high accuracy. In a nutshell, modern tampering techniques and tools go beyond the visuals outpacing the ability of simple visual analysis (spectrogram analysis) to reliably detect tampering[17,18]. This has been verified using an experimental editing function performed on a digital audio file in this study, wherein a cut manipulation (silence to silence editing and in phase editing) was performed precisely on an original unaltered digital audio file using software audacity. The spectrographic and waveform analysis did not reveal any discontinuities or visual telltales at the position of manipulation, thus rendering the standard forgery detection methodology obsolete (refer to Figure2).

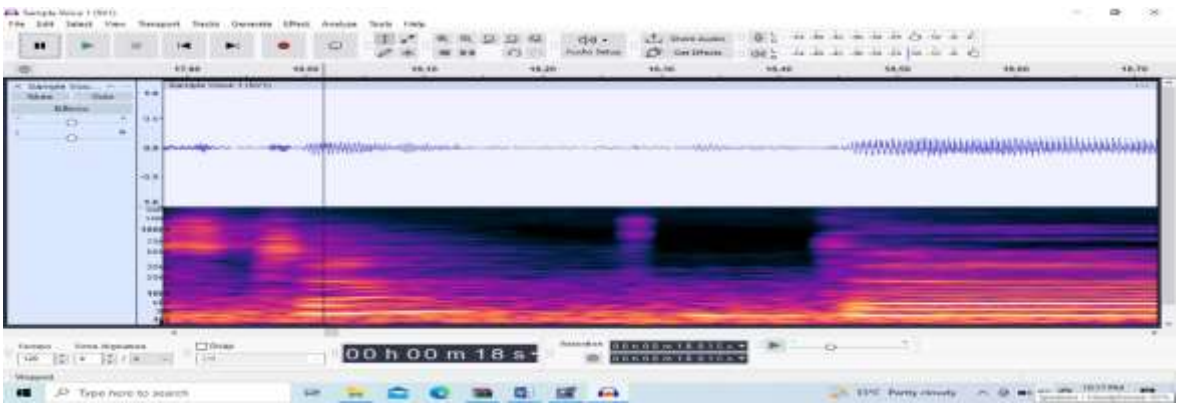

Figure 2: Spectrographic and waveform analysis of the in-phase edited segment of a digital audio file wherein cut manipulation was performed.

The most common practices performed to tamper an audio file include splicing i.e. cutting off a segment of an audio signal from the original audio file, copying a segment of the audio signal and pasting it at some other position in the same audio file or in different audio file, manipulating of audio characteristics like frequency or amplitude of the audio signal or adding or removing background sound or noise, etc to mask the questionable content. This paper presents one such passive detection technique for detection of any tampering in digital audio files. The forgery detection in digital audio files deploy approaches which are broadly classified as Content based approaches and Container based approaches.The approach used in the current study is Container based approach which uses the file structure analysis analysis for detection of forgeries in digital audio files[19].Authors like Koenig and Lacey, Zang et al, Park et al demonstrated that the metadata structure and properties undergo changes when a digital audio file is modified[13,20].

Figure 3: Demonstration of the basic structure of M4A digital audio file

The present study presents one of the passive detection techniques that can identify whether the questioned audio file is original or tampered and processed audio file. The analysis involved in depth analysis of modern multimedia containers like mp4, m4a, mov and related formats. Every such digital audio file is formed of basic structures called Atoms (as per Apple specifications) or Boxes (as per ISO specifications). For in this paper we will use the word atom. Every atom is composed of 4 byte describing size, 4 byte describing type and rest is the data payload which can be of variable size. The basic function of atoms is that these allow streaming of the file, helps in seeking, metadata storage, synchronization and allows editing. The atoms can further contain children atoms. For example a mp4 audio file contain major atoms like- ftyp atom, moov atom, mdat atom, free atom, uuid atom, etc and the atom moov contain children atoms like mvhd, trak, tkhd (refer to Figure-3 which represents the basic structure of a M4A file)[21,22]. The detailed description of each atom and children atoms is given in Figure 4.

The approach discussed in the paper involves the extraction of the information contained in each digital audio file through parsing algorithm and further analysis of multimedia container structure utilizing the parsed data to analyse[8]. The study involves further comparison of the structural values from the parsed data of the original unaltered digital audio files and corresponding tampered digital audio files generated from the existing original audio file.

Figure 4: The parsed data of digital audio files in atomic tree structure form column wise- inbuilt voice recorder original audio file, tampered audio file using GoldWave software, tampered audio file using WavePad software, tampered audio file using Soundop software, tampered audio file using AVS Audio Editor software, original audio file shared via WhatsApp application.

This paper is focused on detection of cutting, copying, copying-pasting and deliberate noise addition type of modifications used to tamper the digital audio files. In a nutshell, this work proposes a comprehensive novel methodology for blind detection of forgery in digital audio files utilizing the MPEG-4 or QuickTime container layout. The methodology discussed in paper relies on the parsing of hierarchical atom structure (parent nodes and nested child nodes) and on exploiting the metadata and the structural values of the container atoms therein to detect tampering. The proposed method not only works to efficiently to identify and distinguish tampered or processed digital audio files but also this method can also be utilized to identify the recording device and the tampering tool used thereof for its modification.

Materials and Methods

The work under the research involved following three phases-

1. The preparation of the data set of original i.e. unaltered digital audio files

2. The preparation of data set of tampered digital audio files, prepared from original (i.e. unaltered) audio files.
3. Forgery detection approach- involving Atomic structure parsing of the digital audio files, extraction of the information contained in each digital audio file with the atom extraction algorithm and multimedia containers structure analysis.

All these phases are discussed in detail below:

Preparation of dataset of Original unaltered digital audio files-

For the preparation of dataset original unaltered digital audio files six different mobile phones were used as a recording device to imitate the real life scenarios. The models selected were- iPhone 15, Samsung Note 8, Samsung S24, Vivo V2348, Motorola MOTO G 62 and OPPO A78 5G. The original samples recorded using these devices were recorded/saved using following modes-

(a) Recordings of inbuilt mobile phone voice recorder (Number of samples- 10x 6=60)
(b) Call recordings auto saved on mobile phone using inbuilt mobile phone call recorder (Number of samples- 10x 6=60)
(c) WhatApp voice notes exchanged using WhatsApp application installed on mobile phone (Number of samples- 10x 6=60)
(d) WhatApp audio files exchanged using WhatsApp application installed on mobile phone (Number of samples- 10x 6=60)

Table 1: Number and Types of digital audio file samples collected from each mobile phone device.

S.No	**Mode of Sample**	**No. of Samples**
1.	**Recordings of inbuilt mobile phone voice recorder**	10 x 6= 60
2.	**Call recordings auto saved on mobile phone using inbuilt mobile phone call recorder**	10 x 6= 60
3.	**WhatApp voice notes exchanged using WhatsApp application installed on mobile phone**	10 x 6= 60
4.	**WhatApp audio files exchanged using WhatsApp application installed on mobile phone**	10 x 6= 60

Following observations were made while preparing the dataset of original unaltered digital audio files (refer to Table-2)-

1. The iPhones don't have an inbuilt call recorder facility.
2. The social media application "WhatsApp" was chosen as the social application to generate samples as it is most widely utilized application used worldwide for exchange of multimedia. The format of WhatsApp voice note was "OPUS" but, the audio files shared as media files through WhatsApp application remain in original file format i.e. same as the format of the file on senders device (the selected formats of shared audio files were MP3 and M4A).
3. A total of 240 original Audio file samples were thus obtained using different devices and methods.
4. Only three types of file formats were observed while analyzing data set of original recordings. These were M4A, MP3 and OPUS file formats.

Table 2: Details of the mobile phone devices used in data preparation and their corresponding file formats of each digital audio file

S.No.	Device	Formats of audio files recorded		
		Inbuilt Voice recorder	Inbuilt Call recorder	WhatsApp voice note
1.	IPhone 15	M4A file	Absent	OPUS file
2.	Samsung Note 8	M4A file	M4A file	OPUS file
3.	Samsung S24	M4A file	M4A file	OPUS file
4.	Vivo V2348	M4A file	M4A file	OPUS file
5.	Motorola MOTO G 62	M4A file	WAV file	OPUS file
6.	Oppo A78 5G	MP3 file	WAV file	OPUS file

Preparation of the data set of tampered digital audio file from original file dataset

The dataset of tampered digital audio files was generated using the dataset of the original digital audio files generated. For the preparation of dataset of tampered digital audio files four different audio editing softwares (GoldWave, WavePad, Soundop, AVS Audio Editor) were used to tamper the original digital audio files dataset. Types of manipulation performed on original samples included cut manipulations, copy-paste manipulations and Noise addition manipulations. Refer to the table 3 for the details of tampering methods used and table 4 for details of tampered audio samples thus generated.

Table 3: Tampering methods used in the study

S.No.	Type of Tampering	Methodology
1.	Cut manipulations	i.Removal of a temporal segment from the beginning of the original file. i.Removal of a temporal segment from the end of the original file. i.Removal of a temporal segment from the middle portion of the original file.
2.	Copy-paste manipulations	i.Copying a segment of original audio file and pasting it in same audio file at some other position. i.Copying a segment of original audio file and pasting it in another original audio file.
3.	Noise addition manipulations	i.Changes (enhancement) in the noise floor of the background noise causing obstruction in the original audio file signal. i.Deliberate noise addition (by layering) in the background of the original audio file.

Table 4: Dataset of original audio files and tampered audio files in detail

	Original unaltered audio files				**Tampered Audio Files**		
	inbuilt mobile phone voice recorders	inbuilt mobile phone call recorders	whatsApp shared original audio files	whatsApp shared voice notes	cut manipulations	copy-paste manipulations	Noise addition manipulations
iPhone 15	10	10	10	10	20+20+20	20+20	20+20
Samsung S24	10	10	10	10	20+20+20	20+20	20+20
Samsung Note 8	10	10	10	10	20+20+20	20+20	20+20
Vivo V2348	10	10	10	10	20+20+20	20+20	20+20
Motorola MOTO G 62	10	10	10	10	20+20+20	20+20	20+20
Oppo A78 5G	10	10	10	10	20+20+20	20+20	20+20
TOTAL	60	60	60	60	360	240	240
	TOTAL =240				**TOTAL =840**		

The Forgery Detection Approach

Container Structure Parsing and analysis-

Metadata i.e. data about data is of three types- descriptive, structural and administrative metadata. The structural metadata of the multimedia file describes containers of data. Thus, Multimedia file containers are a type of metadata. A digital audio file container can be described as a structured envelope that organizes and stores the core components of a digital audio file—namely, the encoded audio data (created by a codec), associated metadata (like track information), and the supplementary data to synchronise different streams. The different formats of digital audio files that were encountered during the current study were- M4A, OPUS and MP3 files. The container structure parsing and analysis of each of these original files and henceforth of the tampered files was thus performed to differentiate between them[2,20].

In case of M4A files the digital audio file container was found to be built using little blocks called atoms which are hierarchical in nature. Each atom has a name or type, which is a special four-letter ASCII code (like 'moov' or 'trak'), usually written in lowercase. Very interesting fact about these Container atoms is that they follow a similar structure for a particular type of file created or processed using particular software or application. In this paper the algorithm that has been used is capable of parsing and analysing information from M4A and MP4 digital audio file container atoms[2] (Figure- 5). The analysis involved parsing the file to reveal the hierarchy of atoms (e.g., ftyp, moov, trak, mdia, minf, stbl, mdat). The files were parsed into their constituent atoms (boxes), revealing the nested hierarchy of metadata and media data and then mapped parent–child relationships between atoms, thus, the internal organization of the file was reconstructed and analyzed. The details of major atom and children atoms present in container based digital audio files are represented in Table 5.

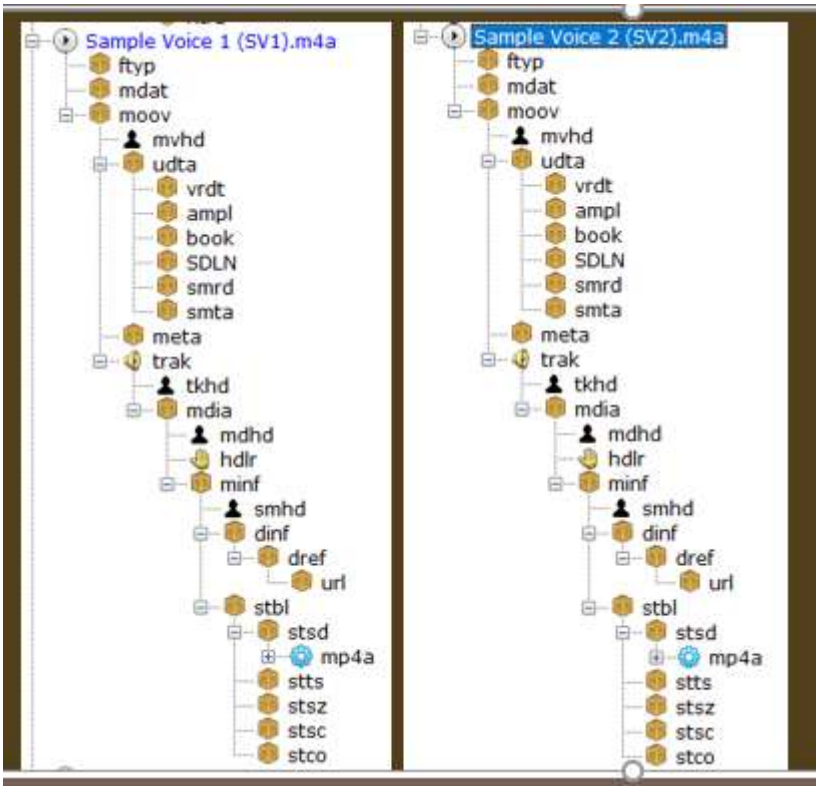

Figure:5- Digital audio file container atoms and children atoms analysis of original audio recording recorded using inbuilt voice recorder of mobile phone-Samsung Note8 (left) and original recording shared via WhatsApp application of same phone.

In the case of OPUS files, the scenario is a bit different, Opus itself is a codec. When stored as .opus or .ogg, the OGG container provides page/packet structures and not the blocks called atoms. OPUS files, thus, doesn't have that kind of nested atom structure that MP4 or M4A files display. OPUS files have Ogg container with pages that hold packets. When parsed through using the python script the OPUS files show container-based page and packet structure[23]. Refer to the Figure 6 for the general anatomy of opus files. It is this hierarchichal packet structure that has been used in this study to distinguish between original unaltered opus digital audio files from the tampered opus digital audio files. We analyzed Opus audio files at the structural level using a combination of parsing tools. The container type (Ogg or WebM) was identified first. We then employed opusinfo to extract codec headers and metadata, oggDump to examine Ogg page boundaries, and ffprobe to export a JSON hierarchy of packets, streams, and tags. This allowed us to reconstruct the internal structure of the file, including headers (OpusHead), metadata (OpusTags), and audio packets. The hierarchical breakdown provided evidence of encoding parameters and potential editing or re-encoding.

MP3 files also have a defined internal structure (see Figure-7), though it's much simpler than Ogg or MP4/M4A files. MP3 files are not container-based but are a raw stream of sequence of independent frames. Each frame has a header (bitrate, sample rate, channel mode) and audio data. Parsing of a mp3 file scans frame headers sequentially and not atoms[24,25]. In case of MP3 files again ffmpeg and ffprobe were utilised to parse the original and tampered files.

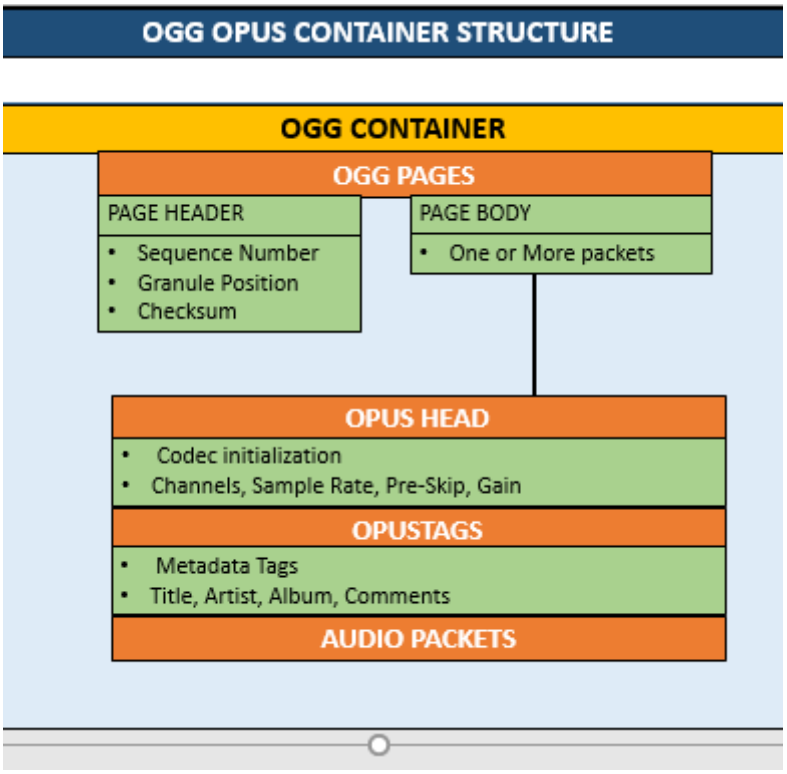

Figure 6 : The core structural hierarchy of an Opus file inside an Ogg container

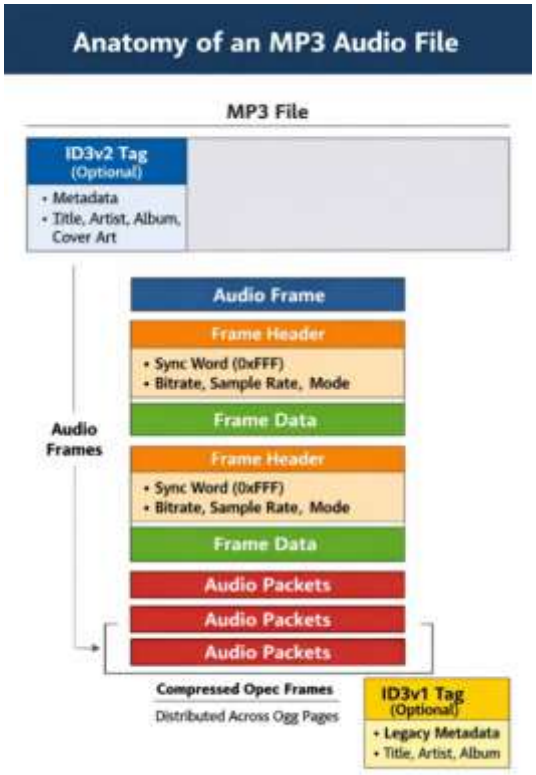

Figure 7: The core structure of a MP3 file

Table 5: Details of major atom and children atoms present in container based digital audio files

S.No.	**Atom/Children atom (#)**	**Full Form**	**Function**
1.	ftyp	File Type atom	Identifies file format, compatible brands, version
2.	moov	Movie atom	Main metadata container. It is the brain of the file. It contains indexes and playback metadata

3.	mdat	Media Data atom	Contains actual compressed audio/video samples
4.	free	Free Space atom	Padding space for future editing
5.	skip	Skip atom	Ignored filler space
6.	meta	Metadata atom	Metadata container
7.	moof	Movie Fragment atom	Used in fragmented MP4 streaming
8.	mfra	Movie Fragment Random Access	Random access info for fragments
9.	pdin	Progressive Download Information	Streaming/download optimization
10.	wide	Wide atom	Alignment/padding atom
11.	#mvhd	Movie Header atom	Global movie information: duration, timescale, creation time, playback rate
12.	#trak	Track atom	Represents one media stream (audio, video, subtitles)
13.	#udta	User Data atom	User/application metadata
14.	#iods	Object Descriptor atom	MPEG object descriptors and stream info
15.	#mvex	Movie Extends atom	Used in fragmented MP4 streaming
16.	#tref	Track Reference atom	References related tracks
17.	#edts	Edit atom	Timeline edits and synchronization
18.	#eist	Edit List atom	Playback timing edits
19.	#mdia	Media atom	Media-specific metadata
20.	#mdhd	Media Header atom	Media timescale and duration
21.	#hdlr	Handler Reference atom	Declares media type (sound, video, text)
22.	#minf	Media Information atom	Holds sample and codec information
23.	#smhd	Sound Media Header atom	Audio playback settings (balance etc.)
24.	#vmhd	Video Media Header atom	Video rendering info
25.	#nmhd	Null Media Header atom	Used for non-audio/video tracks
26.	#dinf	Data Reference atom	References media source
27.	#stbl	Sample Table atom	Maps samples to timing and storage
28.	#stsd	Sample Description atom	Codec information and decoder configuration
29.	#stss	Time-To-Sample atom	Maps sample count to playback time
30.	#ctts	Composition Time-To-Sample atom	Presentation timing offsets
31.	#stsc	Sample-To-Chunk Box	Maps samples into chunks
32.	#stsz	Sample Size atom	Size of every sample
33.	#stco	Chunk Offset atom	32-bit offsets to media chunks
34.	#co64	64-bit Chunk Offset atom	Large-file version of #stco
35.	#stss	Sync Sample atom	Keyframe indexes
36.	#sdtp	Sample Dependency Type atom	Dependency relationships between frames
37.	#padb	Padding Bits atom	Padding info for samples
38.	#meta	Metadata atom	General metadata container
39.	#ilst	Item List Box	iTunes-style metadata tags

Results and Discussion

The analysis of all three types of files i.e. M4A, OPUS and MP3 files was thus performed as per the research methodology discussed above. The results for each type is discussed below-

M4A Files-

1. The container atom analysis performed for each of the original audio files generated using inbuilt voice recorder revealed the same hierarchical pattern (refer to Figure-4) for original unaltered files generated using all the mobile phone devices.
2. The container atom analysis performed for each of the original audio files processed using WhatsApp application revealed the same hierarchical pattern (refer to Figure-4) for original unaltered voice notes generated using all the mobile phone devices. However, the pattern was different from those observed in original audio files generated using inbuilt voice recorder application of each mobile phone.
3. The tampered audio files generated by manipulating the original audio files using different softwares (GoldWave, WavePad, Soundop and AVS Audio Editor) were analysed. It was observed that the base hierarchical pattern of the original files was disrupted. The disruption occurred either due to addition of

additional children atom like 'uuid', 'beam', 'free' or due to the change in position of the container atom like 'mdat', 'moov', 'udta', etc (Figure-4).

4. Each audio editor (GoldWave, WavePad , Soundop, AVS Audio Editor) has a unique pattern called Signature Atomic Structure (Figure-4). The major breakthrough of the analysis was that one type of software produced particular hierarchical pattern of container atoms. Thus, the proposed technique was not only able to distinguish the tampered file from the original file but also it could differentiate and identify different softwares/applications used to tamper or process the file. Refer to the Figure-4 for comparison of results obtained for original files and tampered and processed files.

OPUS Files

The Opus codec, encapsulated in Ogg files, is the default format used by WhatsApp to encode and share voice notes. This format is optimized for speech compression, making it efficient for mobile messaging applications. The dumps of FFprobe of original file (i.e. audio file shared via WhatsApp) and tampered files (edited after being shared through WhatsApp) were compared and analyed. The eiting tools used were GoldWave, WavePad and SoundOp tools. The AVS audio editor tool did not support opus files. The major changes observed were as follows-

The tampered file was re-encoded into smaller frame sizes (original file had- 120 ms per frame), which strongly signaled re-encoding.

Smaller packet sizes of tampered file were found consistent with re-encoding at a lower bitrate or different frame configuration.

Original file: Packet sizes are relatively large (200–335 bytes) whereas

Edited file: Packet sizes are much smaller (30–60 bytes) (refer to Figure-8)

Major difference seen between original and tampered file was the additional [SIDE_DATA] block (refer to Figure-9 and 10). This block is not the usual part of the original Ogg/Opus packet structure. It's extra information which gets embedded when the encoder adds padding or trimming instructions.

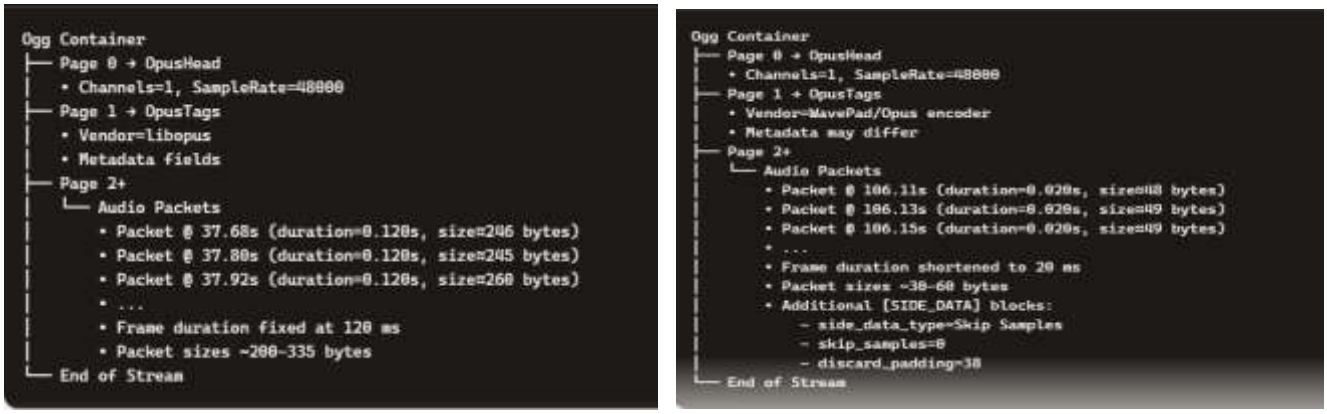

Figure 8: The parsed data of original digital audio files (left) vs tampered digital audio file (right) seen upon container structure analysis.

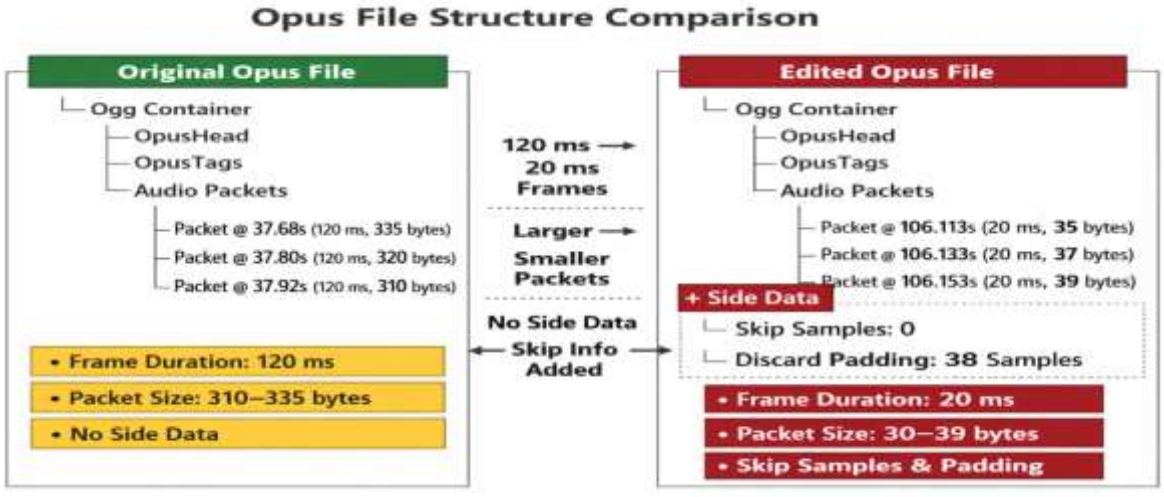

Figure 9: The left side window is container structure analysis of Original unaltered digital audio file and right side window shows the container structure analysis of tampered file

MP3 Files

Since MP3 files are not container-based but are a raw stream of sequence of independent frames, the parsing data obtained for original and tampered MP3 files showed no disconitnuities or additional information encoded in tampered digital audio files. Some tampered MP3 Files showed editing indicators in metadata like encoder signatures in hex information and a fixed packet and frame size. However, a fixed criterion could not be detected for tamper detection in MP3 files.

1. WhatsApp Audio Files need a special mention, the container-structure parsing of WhatsApp shared M4A audio files revealed similar trends. The WhatsApp processed audio files maintained their hierarchical pattern. These WhatsApp processed files when edited using audio editing softwares used in the current study revealed the change in the original hierarchical pattern of various atoms and children atoms like udta trak, etc and also showed presence of additional atoms and children atoms in the structure using padding atoms like uuid, Xtra,etc. in addition to elimination of certain children atoms like some children atoms of udta.
2. Another special mention required is that each mobile phone device used in the study had its own different signature udta atom structure. This unique signature atom structure of udta can therefore also can be used to distinguish or compare the audio file data of different recording mobile phone devices. The udta in case of Motorola device used in the study revealed elimination of vrdt, metd, ampl, book children atoms whereas vivo had these in addition to the basic SDLN, smrd, and smta children atoms.
3. The ISO Base Media File Format permits structural flexibility, deliberate editing combined with meticulous re-indexing (i.e., updating stco/co64 sample offset tables) allows tampered audio files to present a structurally valid, fully compliant atom hierarchy. It is extremely easy to execute, but surprisingly difficult to do cleanly without leaving technical footprints. Modern NLEs (Non-Linear Editors) and audio softwares leave obvious software footprints. They alter the ftyp compatible brands, add unique metadata tags (e.g., lavf tags or encoder strings), and write the atom tree in a specific layout that reveals which software created the file. Even if the manipulator does target specific re-muxing to hide the edit from high-level tree parsers like ffprobe subtle discrepancies (like slight differences in padding atom sizes or vendor-specific metadata tags) often remain.
4. Within the criminal justice framework, the critical utility of this forensic parsing technique rests in its capacity to delineate primary from secondary digital audio evidence. Admissibility and evidentiary weight often hinge on whether a file is an unadulterated original or a secondary re-muxed copy. By exposing subtle container-level anomalies and offset manipulation, this technique equips forensic examiners with objective criteria to validate file provenance, thereby preventing manipulated or secondary audio from being mischaracterized as authentic primary evidence in court proceedings.

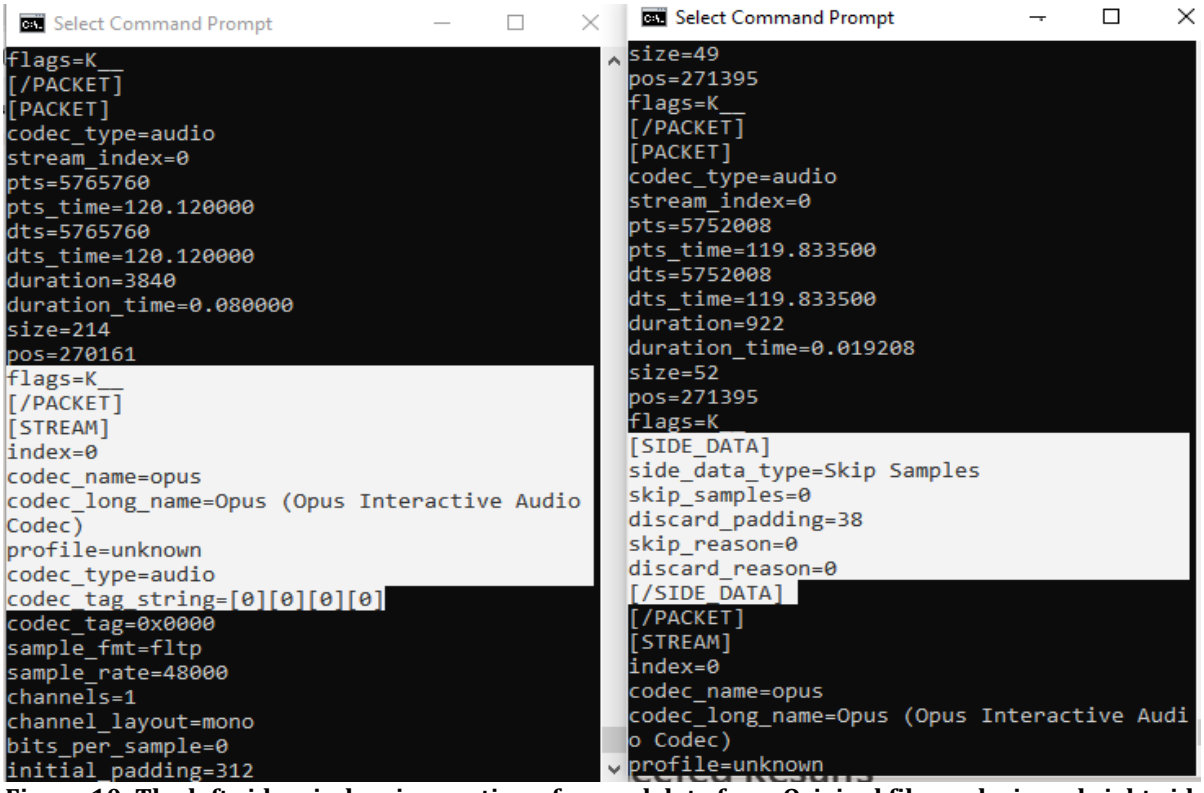

Figure 10: The left side window is a section of parsed data from Original file analysis and right side window shows the comparative section of parsed data from tampered file analysis

4. Conclusions

The blind detection technique i.e. **the Container-Structure Parsing and analysis** technique proposed in this research paper utilizes the atom extraction algorithm,FFmpeg, FFprobe and Multimedia containers structure analysis technique and capitalizes on the fact that any manipulation performed on the digital audio files (like cutting, copy-move, background noise addition, etc) will leave encoder footprints, non contiguous chunk offsets, padding artifacts, sample count discontinuities and thus overall alter the hierarchal container structure of a digital audio files like- M4A and OPUS files. Thus, the **Container Structure Parsing and analysis** technique is an easy, fast and a reliable method as a passive forgery detection technique for digital audio files- M4A and OPUS files. Not only this technique is a strong forensic indicator of **editing or transcoding** to distinguish between original and tampered M4A and OPUS audio file but it can also be used as a promising tool to identify the type of the editing software used for tampering the M4A file and localise the tamper position in case of OPUS files.The greatest strength of this technique is that it is a passive technique and would not require the original reference audio file or recording device for comparison or analysis. The ultimate practical application of this parsing framework lies in its capacity to assist forensic experts and judicial authorities in classifying audio files as primary versus secondary evidence. The current study is limited to six devices and four audio editing softwares which can be extended further to more devices and software tools in future. Also future plan holds to compare similar parsing data of audio files of different models of mobile phone devices belonging to the same parent brand.

Conflict of Interest

"The authors declare no conflict of interest.".

References

[1] M. A. Nasr, W. El-Shafai, N. Abdel-Salam, E.-S. M. El-Rabaie, A. S. El-Fishawy, and F. E. A. El-Samie, "A comprehensive survey of audio forgery detection: challenges and novel trends," Journal of Electrical Systems and Information Technology, vol. 12, no. 1, Jun. 2025, doi: 10.1186/s43067-025-00225-w.

[2] H. Son, S. W. Beak, and J. W. Park, "Automated Detection of Container-based Audio Forgery Using Mobile Crowdsourcing for Dataset Building," APTISI Transactions on Technopreneurship, vol. 6, no. 1, pp. 119–135, Mar. 2024, doi: 10.34306/att.v6i1.383.

[3] M. A. Nasr, W. El-Shafai, N. Abdel-Salam, E.-S. M. El-Rabaie, A. S. El-Fishawy, and F. E. A. El-Samie, "A comprehensive survey of audio forgery detection: challenges and novel trends," Journal of Electrical Systems and Information Technology, vol. 12, no. 1, Jun. 2025, doi: 10.1186/s43067-025-00225-w.

[4] C. Pan et al., "Audio Editing in the Era of Foundation Models: A Survey," Jun. 2026, [Online]. Available: http://arxiv.org/abs/2606.23139

[5] W. Jackson, Digital Audio Editing Fundamentals.

[6] P. R. Bevinamarad and M. S. Shirldonkar, "Audio Forgery Detection Techniques: Present and Past Review," in Proceedings of the 4th International Conference on Trends in Electronics and Informatics, ICOEI 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 613–618. doi: 10.1109/ICOEI48184.2020.9143014.

[7] M. Steinebach, "Watermarking-Based Digital Audio," pp. 1001–1015, 2003.

[8] M. Botta, D. Cavagnino, and A. Marra, "Audio Signal Authentication and Integrity Protection," Computers, vol. 15, no. 4, Apr. 2026, doi: 10.3390/computers15040246.

[9] R. Yang, Z. Qu, and J. Huang, "Detecting digital audio forgeries by checking frame offsets," MM and Sec'08: Proceedings of the 10th ACM Workshop on Multimedia and Security, pp. 21–26, 2008, doi: 10.1145/1411328.1411334.

[10] M. Imran, Z. Ali, S. T. Bakhsh, and S. Akram, "Blind Detection of Copy-Move Forgery in Digital Audio Forensics," IEEE Access, vol. 5, no. c, pp. 12843–12855, 2017, doi: 10.1109/ACCESS.2017.2717842.

[11] M. Zakariah, M. K. Khan, and H. Malik, "Digital multimedia audio forensics: past, present and future," Multimed. Tools Appl., vol. 77, no. 1, pp. 1009–1040, 2018, doi: 10.1007/s11042-016-4277-2.

[12] J. Kaur, M. Joshi, S. Sharma, and T. Sharma, "Detection of variations in temporal frequency and pitch of voice for spatial audio forgery detection," Sep. 2025, p. 40005. doi: 10.1063/5.0272797.

[13] B. E. Koenig and D. S. Lacey, "Audio and Video Files," 1974.

[14] B. Ustubioglu, G. Tahaoglu, G. Ulutas, A. Ustubioglu, and M. Kilic, "Audio forgery detection and localization with super resolution spectrogram and keypoint based clustering approach," Feb. 06, 2023. doi: 10.21203/rs.3.rs-2534047/v1.

[15] A. Ghasemzadeh and E. Esmaeili, "A novel method in audio message encryption based on a mixture of chaos function," Int. J. Speech Technol., vol. 20, no. 4, pp. 829–837, 2017, doi: 10.1007/s10772-

017-9452-y.
[16] P. Indore, "Comprehensive Deep Learning Framework of Image, Audio and Video Forgery Detection," Int. J. Res. Appl. Sci. Eng. Technol., vol. 13, no. 1, pp. 1876–1879, Jan. 2025, doi: 10.22214/ijraset.2025.66641.
[17] R. C. Maher, "Audio Forensic Examination: Authenticity, enhancement, and interpretation," Ieee Signal Processing Magazine [84], no. March, pp. 84–94, 2009.
[18] A. Ashish Sonawane, K. Kishor Turankar, A. Anil Chavan, and A. Taskar, "AudiForgery Detection Using LFCC and Deep Neural Networks." [Online]. Available: www.ijfmr.com
[19] P. S. Marathe, G. C. Wayal, V. S. Pawade, and S. V Ghumatkar, "METADATA AND CONTAINER STRUCTURE ANALYSIS FOR AUDIO AUTHENTICATION," 2023. [Online]. Available: http://www.ijeast.com
[20] R. Ramos Lopez, E. Almaraz Luengo, A. L. Sandoval Orozco, and L. J. G. Villalba, "Digital video source identification based on container's structure analysis," IEEE Access, vol. 8, pp. 36363–36375, 2020, doi: 10.1109/ACCESS.2020.2971785.
[21] Z. Xiang, J. Horváth, S. Baireddy, P. Bestagini, S. Tubaro, and E. J. Delp, "Forensic Analysis of Video Files Using Metadata," Jan. 2022, doi: 10.1109/CVPRW53098.2021.00115.
[22] L. Zhao and L. Guan, "An optimized method and implementation for parsing MP4 metadata," in 2010 IEEE International Conference on Progress in Informatics and Computing, Dec. 2010, pp. 984–987. doi: 10.1109/PIC.2010.5687866.
[23] "ietf.org_rfc_rfc6716.txt".
[24] A. G. Boyarov, I. S. Siparov, А. Г. Бояров, and И. С. Сипаров, "Теория и практика судебной экспертизы Том 14, № 4 (2019) Theory and Practice of Forensic Science," vol. 14, no. 4, pp. 125–136, 2019, doi: 10.30764/1819-2785-2019-14-4-125-136.
[25] S. J and K. P, "Detecting the Hidden: A Comprehensive Review of MP3 Steganalysis Techniques," May 06, 2025. doi: 10.21203/rs.3.rs-6588907/v1.