INVESTIGATING THE STREAM HASHING FUNCTION IN FFMPEG

by

JOHN WOODWARD SMYLY

B.A., Media Communications, University of Michigan, 2001

A thesis submitted to the

Faculty of the College of Arts & Media of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Media Forensics Program

2023

This thesis for the Master of Science degree by

John Woodward Smyly

has been approved for the

Media Forensics Program

by

Catalin Grigoras, Chair

Cole Whitecotton

Gregory Wales

Date: December 16, 2023

Smyly, John Woodward (M.S., Media Forensics Program)

Investigating the Stream Hashing Function in FFmpeg

Thesis directed by Associate Professor Catalin Grigoras

ABSTRACT

As the saying goes, you shouldn't throw the baby out with the bathwater. In other words, make distinctions between items of value and items without value, and treat them accordingly. Interestingly enough, this concept reaches beyond the realm of daily living and can be directly applied to the discipline of digital multimedia forensics, specifically, as it relates to the authentication of data through the process of stream hashing. Traditionally, file hashing has been a core practice in the field of digital forensics and is used in the comparison of files for the purpose of determining if they have been altered. This can ultimately result in the entire data of a given file being dismissed as evidence in a courtroom setting. More recently, it has been demonstrated that stream hashing is a useful tool that can provide a test of authentication for data bit streams within a file, or for parts of the file. A known universal tool for applying the stream hashing method is FFmpeg (Fast Forward Moving Picture Experts Group). Limited testing has been documented with regard to FFmpeg's stream hashing ability which has shown promising results consistent with other authentication techniques (i.e., carving). Stream hashing provides digital forensic practitioners with a convenient way to verify the authenticity of portions of file data. Even if the file as a whole is not an exact representation of its original, portions of the file known as bit streams may still serve as exact representations of the original file data, and therefore, retain forensic significance.

In effort to further explore FFmpeg's stream hashing functionality, a study of broader varieties of codecs and file containers was undertaken. Stream hashing was applied to the

comparison of the audio bit streams of original files to their transcoded derivatives. This process was carried out in 3 different ways – first, by transcoding original files into derivative files that maintained the codecs of the original audio bit streams but changed the file containers. The second method preserved the original file containers but changed the codec of the audio bit stream. The final method consisted of transcoding the audio bit stream of the original file into a different codec and file container. In each case, once the transcoding was complete, the audio bit stream of the original file was compared to the audio bit stream of the derivative file using FFmpeg's stream hashing function. In addition, these calculations were completed in both FFmpeg version 5.1 and FFmpeg version 6.0, thus, allowing for stream hashing results to be compared across FFmpeg versions.

The results largely supported and expanded on previous findings which indicated that FFmpeg's stream hashing function performed the task of authentication by accurately demonstrating audio bit streams to be the same in original files and derivative files, but also, detecting difference as expected (i.e., PCM WAV files derived from AAC MOV files). However, analysis also indicated that this conclusion appeared to be limited to audio bit stream authentication performed in a single FFmpeg version. More specifically, while audio bit streams of original files and for audio bit streams of their respective derivative files matched one another (within FFmpeg 5.1 or FFmpeg 6.0), their stream hashes differed from one another across versions. Even the same original file with an AAC codec (that had not undergone transcoding) yielded different audio bit stream hash values – one audio bit stream hash value when calculated in FFmpeg 5.1 and a completely different audio bit stream hash value when calculate in FFmpeg 6.0.

Based on the observations documented in this study, stream hashing appears to potentially be a viable method of authentication for audio bit stream data of multimedia files within individual versions of FFmpeg. FFmpeg detected differences between audio bit streams of original files and audio bit streams of transcoded files. FFmpeg also was able to detect when audio bit streams were the same for various codecs housed in a variety of file containers. Conversely, incongruency exists between the stream hashing calculations of FFmpeg 5.1 and FFmpeg 6.0 – audio bit streams targeted from the exact same file yield different results in FFmpeg 5.1 than in FFmpeg 6.0. The explanation for this observation is currently unknown and further testing should be carried out to determine the cause.

The form and content of this abstract are approved. I recommend its publication.

Approved: Catalin Grigoras

DEDICATION

I dedicate this paper first to God, who created me and has graciously gifted me with the ability to engage in cognitive thought. He is the source for all things good in my life, and the reason I have been able to persevere through incredible adversity and learn about Media Forensics, which was previously a mystery to me. Apart from Him, I can literally do nothing.

I also dedicate this paper to my family – Emily, Tyler, and Hayley. One of my greatest purposes in life is to provide for, and care for you, and I press forward because I want the very best for you. I pray that my academic pursuits will lead to good things for our family, now, and in the future.

ACKNOWLEDGEMENTS

I would like to express my thanks to Dr. Catalin Grigoras, Jeff Smith, Cole Whitecotton, Greg Wales and Leah Haloin. In addition to being highly intelligent and capable people, each one of you are also humble, personable, and always willing to help. I truly appreciate all of your assistance and dedication to my success in pursuit of my master’s degree in Media Forensics.

I would also like to thank Paul Taylor for initially directing my attention towards the Media Forensics program at the University of Colorado Denver and casting vision for where it might take me.

TABLE OF CONTENTS

CHAPTER

I. INTRODUCTION 1

Research Problem 1

Stream Hashing and Related Definitions 2

Specific Considerations Surrounding FFmpeg Encoders/Decoders 3

Previous Research 4

II. MATERIALS 5

Data 7

III. METHODOLOGY 16

Methods 16

IV. RESULTS 24

Results 24

V. CONCLUSIONS 56

Future Research 57

REFERENCES 59

APPENDIX 60

LIST OF TABLES

TABLE

1. Original files 7
2. Derivatives of original MOV files transcoded using FFmpeg 5.1 9
3. Derivatives of original MOV files transcoded using FFmpeg 6.0 10
4. Derivatives of original MP4 files transcoded using FFmpeg 5.1 11
5. Derivatives of original MP4 files transcoded using FFmpeg 6.0 12
6. Derivatives of original WAV files transcoded in FFmpeg 5.1 13
7. Derivatives of original WAV files transcoded in FFmpeg 6.0 14
8. Derivatives of original MP3 files transcoded in FFmpeg 5.1 and FFmpeg 6.0 15
9. Audio stream hash of original MOV video files with ACC codec using FFmpeg 5.1 25
10. Audio stream hash of original MOV video files with ACC codec using FFmpeg 6.0 25
11. Audio stream hash of WAV files derived from original MOV files in FFmpeg 5.1 26
12. Audio stream hash of WAV files derived from original MOV files in FFmpeg 6.0 27
13. Audio stream hash of M4A files derived from original MOV files in FFmpeg 5.1 28
14. Audio stream hash of M4A files derived from original MOV files in FFmpeg 6.0 29
15. Audio stream hash of MP4 files derived from original MOV files in FFmpeg 5.1 30
16. Audio stream hash of MP4 files derived from original MOV files in FFmpeg 6.0 31
17. Audio stream hash of original MP4 video files with AAC codec using FFmpeg 5.1 32
18. Audio stream hash of original MP4 video files with AAC codec using FFmpeg 6.0 32
19. Audio stream hash of WAV files derived from original MP4 files in FFmpeg 5.1 33
20. Audio stream hash of WAV files derived from original MP4 files in FFmpeg 6.0 34
21. Audio stream hash of M4A files derived from original MP4 files in FFmpeg 5.1 35

22. Audio stream hash of M4A files derived from original MP4 files in FFmpeg 6.0 36

23. Audio stream hash of MOV files derived from original MP4 files in FFmpeg 5.1.......... 37

24. Audio stream hash of MOV files derived from original MP4 files in FFmpeg 6.0.......... 38

25. Audio stream hash of original WAV audio files with PCM codec using FFmpeg 5.1..... 39

26. Audio stream hash of original WAV audio files with PCM codec using FFmpeg 6.0…..39

27. Audio stream hash of MP3 files derived from original WAV files in FFmpeg 5.1 40

28. Audio stream hash of MP3 files derived from original WAV files in FFmpeg 6.0 41

29. Audio stream hash of M4A files derived from original WAV files in FFmpeg 5.1......... 42

30. Audio stream hash of M4A files derived from original WAV files in FFmpeg 6.0......... 43

31. Audio stream hash of AVI files derived from original WAV files in FFmpeg 5.1 44

32. Audio stream hash of AVI files derived from original WAV files in FFmpeg 6.0 45

33. Audio stream hash of original MP3 audio files using FFmpeg 5.1 46

34. Audio stream hash of original MP3 audio files using FFmpeg 6.0 46

35. Audio stream hash of AVI files derived from original MP3 files in FFmpeg 5.1 47

36. Audio stream hash of AVI files derived from original MP3 files in FFmpeg 5.1 48

37. Stream hashes for group 1 original files and derivatives – FFmpeg 5.1.......................... 50

38. Stream hashes for group 1 original files and derivatives – FFmpeg 6.0.......................... 50

39. Stream hashes for group 2 original files and derivatives – FFmpeg 5.1.......................... 52

40. Stream hashes for group 2 original files and derivatives – FFmpeg 6.0.......................... 52

41. Stream hashes for group 3 original files and derivatives (MP4/MOV) – FFmpeg 5.1..... 53

42. Stream hashes for group 3 original files and derivatives (MP4/MOV) – FFmpeg 6.0..... 54

43. Stream hashes for group 3 original files and derivatives (MP3/WAV) – FFmpeg 5.1 54

44. Stream hashes for group 3 original files and derivatives (MP3/WAV) – FFmpeg 6.0 55

LIST OF FIGURES

FIGURE

1. FFmpeg 5.1 version information... 5

2. FFmpeg 6.0 version information... 6

LIST OF ABBREVIATIONS

Abbreviations	Explanations
AAC	Advance Audio Coding
FFMPEG	Fast Forward Moving Picture Experts Group
NIST	National Institute of Standards and Technology
SWGDE	Scientific Working Group on Digital Evidence

CHAPTER I

INTRODUCTION

Authentication is an essential part of digital forensic analysis and serves to verify that evidence or data has not been changed. The Scientific Working Groups on Digital Evidence (SWGDE) further states that “An audio authentication examination seeks to determine if a recording is consistent with the manner in which it is alleged to have been produced. Consequently, there is no catch-all means of declaring a recording “authentic” without having a clear understanding of what claims the creator holds true about its nature and what specific allegations are being levied against the recording” (SWGDE, 2022, p. 17). In some cases, bad actors may intentionally change data to deceive investigators. It is also possible for the forensic practitioner to unknowingly alter data through the use of trusted hardware and software applications. Regardless of the cause, data, whether intentionally or unintentionally altered, will likely be rendered inadmissible for use in criminal and civil court proceedings if it is not properly preserved. In the world of multimedia forensics there are several applications designed to handle audio, video and image data. It is of crucial importance that these applications be properly understood in their ability to perform tasks in an effective and forensically sound manner. The purpose of this document is to explore the ability of FFmpeg, an open-source tool, to accurately authenticate audio bit stream data as it exists in various original codecs and file containers, and as it exists in transcoded derivatives of original files, and across different versions of FFmpeg.

Research Problem

Is FFmpeg a trustworthy source for obtaining stream hashes for audio bit streams? Does it accurately and consistently calculate stream hash values for data before it is transcoded and after? Do different codecs and file containers impact this process? The early research appears to indicate that the method of stream hashing itself is effective. However, there is a lack of data documenting the process applied to a broader scope of the codecs and file containers in use today. The FFmpeg stream hashing function will be examined through the calculation and comparison of audio bit streams across multiple codecs, file types, and versions of FFmpeg.

Stream Hashing and Related Definitions

As I will discuss in subsequent sections in more detail, the research outlined in this document is largely based on foundational research conducted in Multimedia stream hashing: A forensic method for verification (Wales et al., 2022). In an effort to maintain cohesiveness with previous research, I will be adopting the terminology and definitions used in the previously published paper.

Hash: “Chopping the data file into small pieces and combining them to yield a concise numeric value that can be used to identify the original data file. The alphanumeric value from a hashing function in digital and multimedia forensics may represent the contents of a device, folder, file, or data string. This paper proposes the extension of that concept to multimedia streams within files” (Wales et al. 2022, p. 290).

Multimedia or Media Stream: A bit-stream of single or multiple channel audio of video data. Also, an image transformed int a bit-stream video.” (Wales et al. 2022, p. 290)

“Audio Stream: A bit-stream of single or multiple channel audio data

Image Stream: A bit-stream of an image transformed into a video for rendering output

Video Stream: A bit-stream of single or multiple channel video data” (Wales et al. 2022, p. 290).

Specific Considerations Surrounding FFmpeg Encoders/Decoders

One of FFmpeg's functions is transcoding multimedia files, and therefore, it must be equipped with the correct algorithms for properly encoding audio/video/image data. Specifically, as it relates to the audio focus of this paper, FFmpeg supports three different encoders for dealing with Advanced Audio Codec (AAC) audio data – aac, aac_at, and libfdk_aac (FFmpeg, 2023). All three of these encoders have the capability of handling Advanced Audio Coding Low Complexity (AAC-LC) and two of them, libfdk_aac and aac_at, can also handle High Efficiency Advanced Audio Coding (HE-AAC). The only encoder LIBRARIES and DEFAULT encoders… In this study, it is important to establish an awareness of which FFmpeg encoder is being used to transcode audio data for two reasons:

1) It directly impacts how the audio data in FFmpeg is being transmitted into derivative files, and

2) The accuracy and consistency of the encoder has implications for other software applications that integrate FFmpeg into their functionality; for example, FFmpeg is discussed on Audacity's support portion of its website, and highlights how FFmpeg is required to import and export a variety of audio formats, including M4A and WMA (Audacity, 2023).

Different encoders utilize different syntax for commands. This allows for an important distinction to be made between the various FFmpeg AAC audio encoders and identify which one is being used based on the command syntax. The Native FFmpeg Encoder (AAC) was used to transcode and preserve audio data in this study. In addition to the syntax used, it can be known that the implemented versions of FFmpeg (both 5.1 and 6.0) in this study used the default native AAC encoder based on the fact that no external library was manually enabled via the *--enable-lib option* (FFmpeg, 2023).

Previous Research

A limited amount of research has been dedicated to the application of stream hashing in multimedia forensics. An initial argument has been made for stream hashing being the most advisable way to transcode video and audio streams in the context of video authentication (Wales, 2019). Additionally, a more detailed examination has been applied and used to verify the effectiveness of stream hashing with regard to audio/video/image files (Wales et al., 2022). This paper demonstrated that FFmpeg version 5.1 can effectively calculate stream hashes for audio, images and video, and furthermore, can detect when multimedia streams are different, and when multimedia streams are the same. Regarding audio, the accuracy of stream hashing in FFmpeg was shown to be accurate when applied to PCM WAV files. Additionally, FFmpeg was able to detect differences in an audio stream belonging to a PCM WAV file and the same audio stream after it was transcoded into an MP3 file.

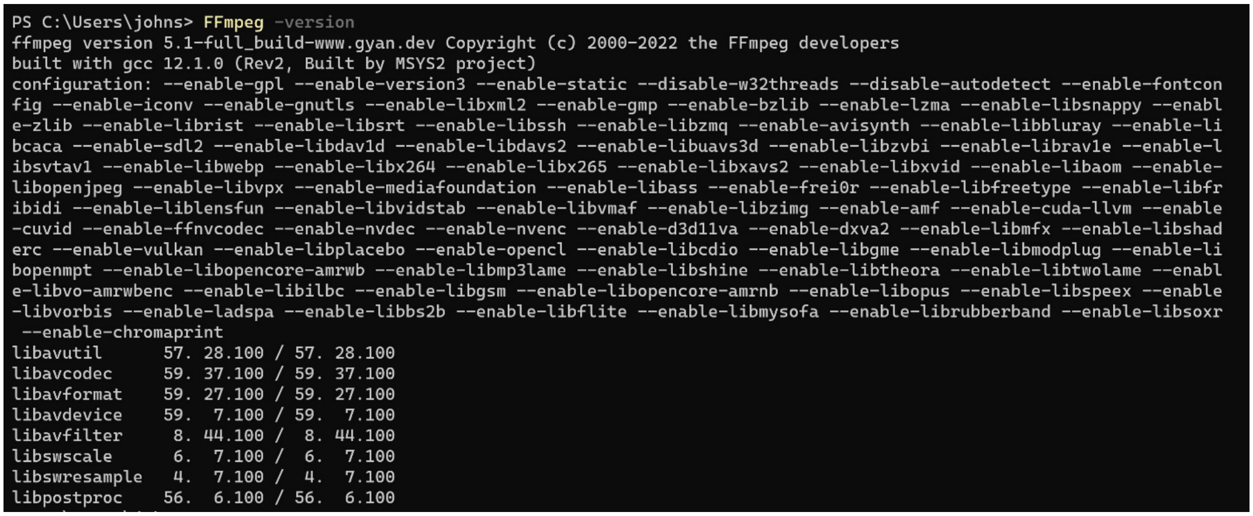
Despite the documented implementation of a more detailed approach toward understanding stream hashing in FFmpeg, there is a lack of research documenting the reliability of stream hashing as it relates to several other codecs and file containers of multimedia files. This paper aims to build on the previous research conducted and offer a deeper exploration into how stream hashing in the FFmpeg open-source application functions as an authentication tool for digital audio streams. Additionally, now that a newer version of FFmpeg (6.0) has been released, it is desirable to expand testing to ensure consistency is up to date across versions.

CHAPTER II

MATERIALS

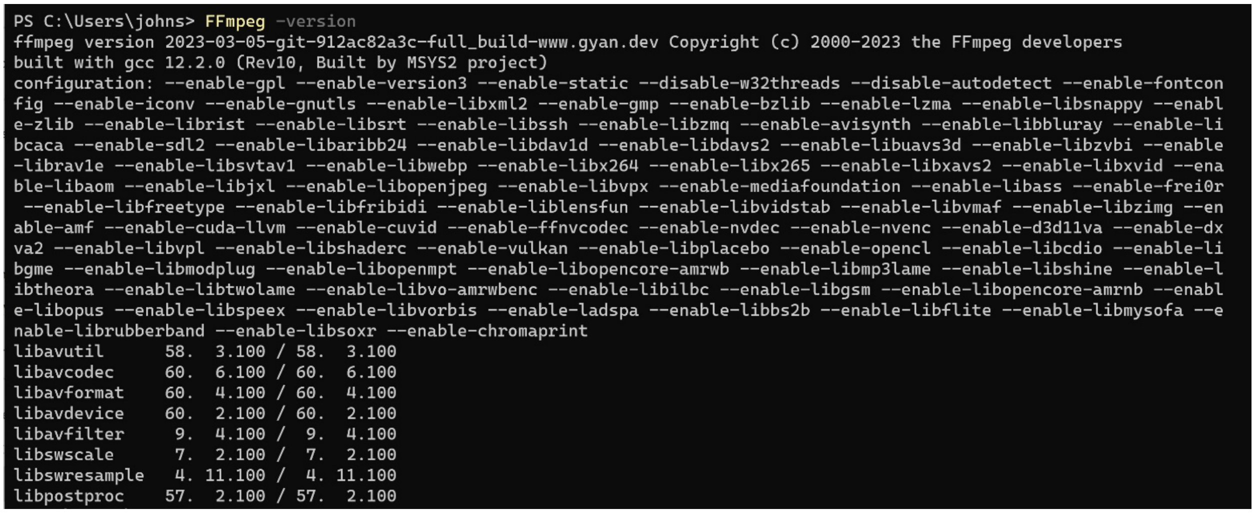
The scope of this study deals with exploring the capability of FFmpeg to: 1) accurately preserve audio streams in multimedia files during the transcoding process, and 2) accurately authenticate audio streams using the stream hashing tool. In addition to these goals, it was important to apply FFmpeg's stream hashing tool on audio data transcoded from video to video, from audio to audio, from video to audio, and from audio to video. These considerations are reflected in the original files as well as in their transcoded derivatives.

FFmpeg version 5.1 and FFmpeg version 6.0 were used to calculate audio stream hashes for this study. Detailed FFmpeg version information was displayed in Windows Powershell by using the FFmpeg command: **FFmpeg -version**. The screen shots of this information can be viewed below:



```
PS C:\Users\johns> FFmpeg -version
ffmpeg version 5.1-full_build-www.gyan.dev Copyright (c) 2000-2022 the FFmpeg developers
built with gcc 12.1.0 (Rev2, Built by MSYS2 project)
configuration: --enable-gpl --enable-version3 --enable-static --disable-w32threads --disable-autodetect --enable-fontcon
fig --enable-iconv --enable-gnutls --enable-libxml2 --enable-gmp --enable-bzlib --enable-lzma --enable-libsnappy --enabl
e-zlib --enable-librist --enable-libsrt --enable-libssh --enable-libzmq --enable-avisynth --enable-libbluray --enable-li
bcaca --enable-sdl2 --enable-libdav1d --enable-libdavs2 --enable-libuavs3d --enable-libzvbi --enable-librav1e --enable-l
ibsvtav1 --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs2 --enable-libxvid --enable-libaom --enable-
libopenjpeg --enable-libvpx --enable-mediafoundation --enable-libass --enable-frei0r --enable-libfreetype --enable-libfr
ibidi --enable-liblensfun --enable-libvidstab --enable-libvmaf --enable-libzimg --enable-amf --enable-cuda-llvm --enable
-cuvid --enable-ffnvcodec --enable-nvdec --enable-nvenc --enable-d3d11va --enable-dxva2 --enable-libmfx --enable-libshad
erc --enable-vulkan --enable-libplacebo --enable-opencl --enable-libcdio --enable-libgme --enable-libmodplug --enable-li
bopenmpt --enable-libopencore-amrwb --enable-libmp3lame --enable-libshine --enable-libtheora --enable-libtwolame --enabl
e-libvo-amrwbenc --enable-libilbc --enable-libgsm --enable-libopencore-amrnb --enable-libopus --enable-libspeex --enable
-libvorbis --enable-ladspa --enable-libbs2b --enable-libflite --enable-libmysofa --enable-librubberband --enable-libsoxr
 --enable-chromaprint
libavutil      57. 28.100 / 57. 28.100
libavcodec     59. 37.100 / 59. 37.100
libavformat    59. 27.100 / 59. 27.100
libavdevice    59.  7.100 / 59.  7.100
libavfilter     8. 44.100 /  8. 44.100
libswscale      6.  7.100 /  6.  7.100
libswresample   4.  7.100 /  4.  7.100
libpostproc    56.  6.100 / 56.  6.100
```

Figure 1: FFmpeg 5.1 version information



```
PS C:\Users\johns> FFmpeg -version
ffmpeg version 2023-03-05-git-912ac82a3c-full_build-www.gyan.dev Copyright (c) 2000-2023 the FFmpeg developers
built with gcc 12.2.0 (Rev10, Built by MSYS2 project)
configuration: --enable-gpl --enable-version3 --enable-static --disable-w32threads --disable-autodetect --enable-fontcon
fig --enable-iconv --enable-gnutls --enable-libxml2 --enable-gmp --enable-bzlib --enable-lzma --enable-libsnappy --enabl
e-zlib --enable-librist --enable-libsrt --enable-libssh --enable-libzmq --enable-avisynth --enable-libbluray --enable-li
bcaca --enable-sdl2 --enable-libaribb24 --enable-libdav1d --enable-libdavs2 --enable-libuavs3d --enable-libzvbi --enable
-librav1e --enable-libsvtav1 --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs2 --enable-libxvid --ena
ble-libaom --enable-libjxl --enable-libopenjpeg --enable-libvpx --enable-mediafoundation --enable-libass --enable-frei0r
 --enable-libfreetype --enable-libfribidi --enable-liblensfun --enable-libvidstab --enable-libvmaf --enable-libzimg --en
able-amf --enable-cuda-llvm --enable-cuvid --enable-ffnvcodec --enable-nvdec --enable-nvenc --enable-d3d11va --enable-dx
va2 --enable-libvpl --enable-libshaderc --enable-vulkan --enable-libplacebo --enable-opencl --enable-libcdio --enable-li
bgme --enable-libmodplug --enable-libopenmpt --enable-libopencore-amrwb --enable-libmp3lame --enable-libshine --enable-l
ibtheora --enable-libtwolame --enable-libvo-amrwbenc --enable-libilbc --enable-libgsm --enable-libopencore-amrnb --enabl
e-libopus --enable-libspeex --enable-libvorbis --enable-ladspa --enable-libbs2b --enable-libflite --enable-libmysofa --e
nable-librubberband --enable-libsoxr --enable-chromaprint
libavutil      58.  3.100 / 58.  3.100
libavcodec     60.  6.100 / 60.  6.100
libavformat    60.  4.100 / 60.  4.100
libavdevice    60.  2.100 / 60.  2.100
libavfilter     9.  4.100 /  9.  4.100
libswscale      7.  2.100 /  7.  2.100
libswresample   4. 11.100 /  4. 11.100
libpostproc    57.  2.100 / 57.  2.100
```

Figure 2: FFmpeg 6.0 version information

I created 40 original files (20 video and 20 audio) to be used in this study. These files also served as the base files from which all other derivative files were made. (note: throughout this document, whenever the word 'original' is used to describe a file, it is referring to one of the files subsequently listed here, meaning a file that was created directly by a software program and not a derivative of another file).

Data

The original files along with the device/software/version used to create them are as follows:

Table 1: Original files

10 original MOV video files with AAC codec using iPhone 13 Pro (iOS version 16.6.1) Camera app	
File #	**File Name**
1	Balloon.MOV
2	Door.MOV
3	Flowers.MOV
4	Rocks.MOV
5	Run.MOV
6	Scooter.MOV
7	Shadow.MOV
8	Sun.MOV
9	Tomato.MOV
10	Trees.MOV

10 original MP4 video files with AAC codec using Lenovo X1 laptop computer (Windows 11) and Xbox Game Bar version 5.823.7272.0	
File #	**File Name**
1	Barry.MP4
2	DirtBike.MP4
3	Horse.MP4
4	Pistons.MP4
5	RedWings.MP4
6	Rollerblades.MP4
7	Skateboard.MP4
8	Surfing.MP4
9	Tigers.MP4
10	Track.MP4

10 original WAV audio files with PCM codec using Lenovo X1 laptop computer (Windows 11) and Audacity version 3.3.3	
File #	**File Name**
1	Music1.wav
2	Music2.wav
3	Music3.wav
4	Music4.wav
5	Music5.wav
6	Music6.wav
7	Music7.wav
8	Music8.wav
9	Music9.wav
10	Music10.wav

10 original MP3 audio files using Lenovo X1 laptop computer (Windows 11) and Audacity version 3.3.3	
File #	**File Name**
1	Song1.mp3
2	Song2.mp3
3	Song3.mp3
4	Song4.mp3
5	Song5.mp3
6	Song6.mp3
7	Song7.mp3
8	Song8.mp3
9	Song9.mp3
10	Song10.mp3

Derivative Files

There was a total of 200 derivative files transcoded from the original files. Some derivative files were created using FFmpeg 5.1 while an equal number were created using FFmpeg 6.0. The naming convention for the derivative transcoded files tracks the original file name, the original file container type, the version of FFmpeg used for transcoding, and lastly, the new file container type.

An example of the naming convention adopted during the transcoding process is illustrated below, referencing the original file, and then displaying derivative files created in both FFmpeg version 5.1 and Fmpeg 6.0.

ORIGINAL FILE

ExampleFile.MOV

DERIVATIVE FILE

ExampleFileMOV_ff5.wav

DERIVATIVE FILE

ExampleFileMOV_ff6.wav

KEY

Green = Original file name

Red = Original file container type

Blue = FFmpeg version used for transcoding

Pink = New file container type

Each of the derivative files are listed in the below tables as follows:

Table 2: Derivatives of original MOV files transcoded using FFmpeg 5.1

WAV files derived from original MOV files in FFmpeg 5.1	
File #	**File Name**
1	BalloonMOV_ff5.wav
2	DoorMOV_ff5.wav
3	FlowersMOV_ff5.wav
4	RocksMOV_ff5.wav
5	RunMOV_ff5.wav
6	ScooterMOV_ff5.wav
7	ShadowMOV_ff5.wav
8	SunMOV_ff5.wav
9	TomatoMOV_ff5.wav
10	TreesMOV_ff5.wav

M4A files derived from original MOV files in FFmpeg 5.1	
File #	**File Name**
1	BalloonMOV_ff5.m4a
2	DoorMOV_ff5.m4a
3	FlowersMOV_ff5.m4a
4	RocksMOV_ff5.m4a
5	RunMOV_ff5.m4a
6	ScooterMOV_ff5.m4a
7	ShadowMOV_ff5.m4a
8	SunMOV_ff5.m4a
9	TomatoMOV_ff5.m4a
10	TreesMOV_ff5.m4a

MP4 files derived from original MOV files in FFmpeg 5.1	
File #	**File Name**
1	BalloonMOV_ff5.mp4
2	DoorMOV_ff5.mp4
3	FlowersMOV_ff5.mp4
4	RocksMOV_ff5.mp4
5	RunMOV_ff5.mp4
6	ScooterMOV_ff5.mp4
7	ShadowMOV_ff5.mp4
8	SunMOV_ff5.mp4
9	TomatoMOV_ff5.mp4
10	TreesMOV_ff5.mp4

Table 3: Derivatives of original MOV files transcoded using FFmpeg 6.0

WAV files derived from original MOV files in FFmpeg 6.0	
File #	**File Name**
1	BalloonMOV_ff6.wav
2	DoorMOV_ff6.wav
3	FlowersMOV_ff6.wav
4	RocksMOV_ff6.wav
5	RunMOV_ff6.wav
6	ScooterMOV_ff6.wav
7	ShadowMOV_ff6.wav
8	SunMOV_ff6.wav
9	TomatoMOV_ff6.wav
10	TreesMOV_ff6.wav

M4A files derived from original MOV files in FFmpeg 6.0	
File #	**File Name**
1	BalloonMOV_ff6.m4a
2	DoorMOV_ff6.m4a
3	FlowersMOV_ff6.m4a
4	RocksMOV_ff6.m4a
5	RunMOV_ff6.m4a
6	ScooterMOV_ff6.m4a
7	ShadowMOV_ff6.m4a
8	SunMOV_ff6.m4a
9	TomatoMOV_ff6.m4a
10	TreesMOV_ff6.m4a

MP4 files derived from original MOV files in FFmpeg 6.0	
File #	**File Name**
1	BalloonMOV_ff6.mp4
2	DoorMOV_ff6.mp4
3	FlowersMOV_ff6.mp4
4	RocksMOV_ff6.mp4
5	RunMOV_ff6.mp4
6	ScooterMOV_ff6.mp4
7	ShadowMOV_ff6.mp4
8	SunMOV_ff6.mp4
9	TomatoMOV_ff6.mp4
10	TreesMOV_ff6.mp4

Table 4: Derivatives of original MP4 files transcoded using FFmpeg 5.1

WAV files derived from original MP4 files in FFmpeg 5.1	
File #	**File Name**
1	BarryMP4_ff5.wav
2	DirtBikeMP4_ff5.wav
3	HorseMP4_ff5.wav
4	PistonsMP4_ff5.wav
5	RedWingsMP4_ff5.wav
6	RollerbladesMP4_ff5.wav
7	SkateboardMP4_ff5.wav
8	SurfingMP4_ff5.wav
9	TigersMP4_ff5.wav
10	TrackMP4_ff5.wav

M4A files derived from original MP4 files in FFmpeg 5.1	
File #	**File Name**
1	BarryMP4_ff5.m4a
2	DirtBikeMP4_ff5.m4a
3	HorseMP4_ff5.m4a
4	PistonsMP4_ff5.m4a
5	RedWingsMP4_ff5.m4a
6	RollerbladesMP4_ff5.m4a
7	SkateboardMP4_ff5.m4a
8	SurfingMP4_ff5.m4a
9	TigersMP4_ff5.m4a
10	TrackMP4_ff5.m4a

MOV files derived from original MP4 files in FFmpeg 5.1	
File #	**File Name**
1	BarryMP4_ff5.MOV
2	DirtBikeMP4_ff5.MOV
3	HorseMP4_ff5.MOV
4	PistonsMP4_ff5.MOV
5	RedWingsMP4_ff5.MOV
6	RollerbladesMP4_ff5.MOV
7	SkateboardMP4_ff5.MOV
8	SurfingMP4_ff5.MOV
9	TigersMP4_ff5.MOV
10	TrackMP4_ff5.MOV

Table 5: Derivatives of original MP4 files transcoded using FFmpeg 6.0

WAV files derived from original MP4 files in FFmpeg 6.0	
File #	**File Name**
1	BarryMP4_ff6.wav
2	DirtBikeMP4_ff6.wav
3	HorseMP4_ff6.wav
4	PistonsMP4_ff6.wav
5	RedWingsMP4_ff6.wav
6	RollerbladesMP4_ff6.wav
7	SkateboardMP4_ff6.wav
8	SurfingMP4_ff6.wav
9	TigersMP4_ff6.wav
10	TrackMP4_ff6.wav

M4A files derived from original MP4 files in FFmpeg 6.0	
File #	**File Name**
1	BarryMP4_ff6.m4a
2	DirtBikeMP4_ff6.m4a
3	HorseMP4_ff6.m4a
4	PistonsMP4_ff6.m4a
5	RedWingsMP4_ff6.m4a
6	RollerbladesMP4_ff6.m4a
7	SkateboardMP4_ff6.m4a
8	SurfingMP4_ff6.m4a
9	TigersMP4_ff6.m4a
10	TrackMP4_ff6.m4a

MOV files derived from original MP4 files in FFmpeg 6.0	
File #	**File Name**
1	BarryMP4_ff6.MOV
2	DirtBikeMP4_ff6.MOV
3	HorseMP4_ff6.MOV
4	PistonsMP4_ff6.MOV
5	RedWingsMP4_ff6.MOV
6	RollerbladesMP4_ff6.MOV
7	SkateboardMP4_ff6.MOV
8	SurfingMP4_ff6.MOV
9	TigersMP4_ff6.MOV
10	TrackMP4_ff6.MOV

Table 6: Derivatives of original WAV files transcoded in FFmpeg 5.1

MP3 files derived from original WAV files in FFmpeg 5.1	
File #	**File Name**
1	Music1wav_ff5.mp3
2	Music2wav_ff5.mp3
3	Music3wav_ff5.mp3
4	Music4wav_ff5.mp3
5	Music5wav_ff5.mp3
6	Music6wav_ff5.mp3
7	Music7wav_ff5.mp3
8	Music8wav_ff5.mp3
9	Music9wav_ff5.mp3
10	Music10wav_ff5.mp3

M4A files derived from original WAV files in FFmpeg 5.1	
File #	**File Name**
1	Music1wav_ff5.m4a
2	Music2wav_ff5.m4a
3	Music3wav_ff5.m4a
4	Music4wav_ff5.m4a
5	Music5wav_ff5.m4a
6	Music6wav_ff5.m4a
7	Music7wav_ff5.m4a
8	Music8wav_ff5.m4a
9	Music9wav_ff5.m4a
10	Music10wav_ff5.m4a

AVI files derived from original WAV files in FFmpeg 5.1	
File #	**File Name**
1	Music1wav_ff5.avi
2	Music2wav_ff5.avi
3	Music3wav_ff5.avi
4	Music4wav_ff5.avi
5	Music5wav_ff5.avi
6	Music6wav_ff5.avi
7	Music7wav_ff5.avi
8	Music8wav_ff5.avi
9	Music9wav_ff5.avi
10	Music10wav_ff5.avi

Table 7: Derivatives of original WAV files transcoded in FFmpeg 6.0

MP3 files derived from original WAV files in FFmpeg 6.0	
File #	**File Name**
1	Music1wav_ff6.mp3
2	Music2wav_ff6.mp3
3	Music3wav_ff6.mp3
4	Music4wav_ff6.mp3
5	Music5wav_ff6.mp3
6	Music6wav_ff6.mp3
7	Music7wav_ff6.mp3
8	Music8wav_ff6.mp3
9	Music9wav_ff6.mp3
10	Music10wav_ff6.mp3

M4A files derived from original WAV files in FFmpeg 6.0	
File #	**File Name**
1	Music1wav_ff6.m4a
2	Music2wav_ff6.m4a
3	Music3wav_ff6.m4a
4	Music4wav_ff6.m4a
5	Music5wav_ff6.m4a
6	Music6wav_ff6.m4a
7	Music7wav_ff6.m4a
8	Music8wav_ff6.m4a
9	Music9wav_ff6.m4a
10	Music10wav_ff6.m4a

AVI files derived from original WAV files in FFmpeg 6.0	
File #	**File Name**
1	Music1wav_ff6.avi
2	Music2wav_ff6.avi
3	Music3wav_ff6.avi
4	Music4wav_ff6.avi
5	Music5wav_ff6.avi
6	Music6wav_ff6.avi
7	Music7wav_ff6.avi
8	Music8wav_ff6.avi
9	Music9wav_ff6.avi
10	Music10wav_ff6.avi

Table 8: Derivatives of original MP3 files transcoded in FFmpeg 5.1 and FFmpeg 6.0

AVI files derived from original MP3 files in FFmpeg 5.1	
File #	**File Name**
1	Song1mp3_ff5.avi
2	Song2mp3_ff5.avi
3	Song3mp3_ff5.avi
4	Song4mp3_ff5.avi
5	Song5mp3_ff5.avi
6	Song6mp3_ff5.avi
7	Song7mp3_ff5.avi
8	Song8mp3_ff5.avi
9	Song9mp3_ff5.avi
10	Song10mp3_ff5.avi

AVI files derived from original MP3 files in FFmpeg 6.0	
File #	**File Name**
1	Song1mp3_ff6.avi
2	Song2mp3_ff6.avi
3	Song3mp3_ff6.avi
4	Song4mp3_ff6.avi
5	Song5mp3_ff6.avi
6	Song6mp3_ff6.avi
7	Song7mp3_ff6.avi
8	Song8mp3_ff6.avi
9	Song9mp3_ff6.avi
10	Song10mp3_ff6.avi

CHAPTER III

METHODOLOGY

The experimental design for this study was centered around the creation of original digital audio and video files, calculating stream hashes of audio data in the original files, transcoding the original files, calculating the stream hashes of audio data in the transcoded/derivative files, and lastly, comparing the audio stream hash values of the original files with the audio stream hash values of the transcoded files. This process took place within individual versions of FFmpeg 5.1 and 6.0 as well as across these different versions. Similarities and differences between audio stream hash values were recorded. All audio stream hashes were calculated using the SHA256 algorithm which is the default hash algorithm for FFmpeg (FFmpeg, 2023). The SHA256 algorithm was also chosen due it meeting minimum national standards for hash algorithms in the digital forensic sciences (NIST, 2023). FFprobe was also used on each original file as a recommended method of devising the numbers of audio streams present (SWGDE, 2018). The first audio stream listed in each file was used for the calculation of the audio bit stream hash value.

Methods

First, original files were first created. The 10 original MOV video files were made using an iPhone 13 Pro with iOS version 16.6.1. The 10 original MP4 files were made using Xbox Game Bar version 5.823.7272.0 on a Lenovo Thinkpad X1 Extreme laptop computer with a Windows 11 operating system. The 10 original WAV files and 10 original MP3 files were made using Audacity version 3.3.3.

Next, hash values of audio bit streams for each original file were calculated in FFmpeg 5.1 and recorded. Hash values of audio bit streams for each original file were calculated in FFmpeg 6.0 as well and recorded.

Once stream hashes of audio bit streams for the original files had been recorded, the original files were subjected to FFmpeg commands and transcoded into various codecs and file container types.

Following the creation of derivative files, hash values for each the audio bit streams of contained within them were calculated and compared to the audio bit streams previously recorded for their original file counterparts. This comparison was done within the confines of each version of FFmpeg, and also, across versions.

The specific steps taken for creating and comparing each original and derivative file set is outlined below and organized according to the original file that a given derivative file was transcoded from. It documents the process for each file and its derivatives in terms of creation, stream hash calculation and comparison, FFmpeg versions used, and the specific FFmpeg command used for each step.

FFmpeg 5.1

Audio stream hash values were obtained and recorded for each original MOV file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mov -map 0:a:0 -f streamhash –
```

1a.

Audio streams were bifurcated from each original MOV file and transcoded into audio files with a PCM codec and a WAV container using the following command:

```
ffmpeg -i filename.mov -vn -acodec pcm_s16le -ar 44100 -ac 1 filename.wav
```

Audio stream hash values were obtained and recorded for each bifurcated/transcoded WAV file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.wav -map 0:a:0 -f streamhash –
```

Comparisons between audio hashes for original MOV files and PCM WAV files derived from original MOV files were observed and recorded.

1b.

Audio streams were bifurcated from each original MOV file and transcoded into audio files with an AAC codec and a M4A container using the following command:

```
ffmpeg -i filename.mov -vn -acodec pcm_s16le -ar 44100 -ac 1 filename.m4a
```

Audio stream hash values were obtained and recorded for each bifurcated/transcoded M4A file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.m4a -map 0:a:0 -f streamhash –
```

Comparisons between audio stream hashes for original MOV files and PCM WAV files derived from original MOV files were observed and recorded.

Audio stream hash values were obtained and recorded for each original MP4 file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mov -map 0:a:0 -f streamhash –
```

1c.

Audio stream hash values were obtained and recorded for each original MP4 file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mp4 -map 0:a:0 -f streamhash –
```

Audio streams were bifurcated from each original MP4 file and transcoded into audio files with a PCM codec and a WAV container using the following command:

```
ffmpeg -i filename.mp4 -vn -acodec pcm_s16le -ar 44100 -ac 1 filename.wav
```

Audio stream hash values were obtained and recorded for each bifurcated/transcoded WAV file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.wav -map 0:a:0 -f streamhash –
```

Comparisons between audio hashes for original MOV files and PCM WAV files derived from original MP4 files were observed and recorded.

1d.

Audio streams were bifurcated from each original MP4 file and transcoded into audio files with an AAC codec and a M4A container using the following command:

```
ffmpeg -i filename.mp4 -vn -acodec pcm_s16le -ar 44100 -ac 1 filename.m4a
```

Audio stream hash values were obtained and recorded for each bifurcated/transcoded M4A file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.m4a -map 0:a:0 -f streamhash –
```

****Each of these steps was repeated in FFmpeg 6.0****

Audio stream hash values were obtained and recorded for each original WAV file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mov -map 0:a:0 -f streamhash –
```

2a.

Audio streams from each original WAV file were transcoded into audio files with an MP3 codec and a MP3 container using the following command:

```
ffmpeg -i filename.wav -acodec mp3 filename.mp3
```

Audio stream hash values were obtained and recorded for each transcoded MP3 file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mp3 -map 0:a:0 -f streamhash –
```

2b.

Audio streams from each original WAV file were transcoded into audio files with an AAC codec and a M4A container using the following command:

```
ffmpeg -i filename.wav -acodec aac filename.mp3
```

Audio stream hash values were obtained and recorded for each transcoded M4A file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.m4a -map 0:a:0 -f streamhash –
```

****Each of these steps was repeated in FFmpeg 6.0****

Audio stream hash values were obtained and recorded for each original MP3 file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mp3 -map 0:a:0 -f streamhash –
```

3a.

Audio streams from each original MP3 file were transcoded into AVI files with the same MP3 codec and an AVI container using the following command:

```
ffmpeg -i filename.mp3 -c copy filename.avi
```

Audio stream hash values were obtained and recorded for each transcoded AVI file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.avi -map 0:a:0 -f streamhash –
```

Comparisons between audio hashes for original MP3 files and AVI files derived from original MP3 files were observed and recorded.

3b.

Audio streams from each original WAV file were transcoded into AVI files with the same MP3 codec and an AVI container using the following command:

```
ffmpeg -i filename.mp3 -c copy filename.avi
```

Audio stream hash values were obtained and recorded for each transcoded M4A file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.avi -map 0:a:0 -f streamhash –
```

Comparisons between audio hashes for original WAV files and AVI files derived from original WAV files were observed and recorded.

3c.

Audio streams from each original MOV file were transcoded into MP4 files with the same AAC codec and an MP4 container using the following command:

```
ffmpeg -i filename.mov -c copy filename.mp4
```

Audio stream hash values were obtained and recorded for each transcoded M4A file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mp4 -map 0:a:0 -f streamhash –
```

Comparisons between audio stream hashes for original MOV files and MP4 files derived from original MOV files were observed and recorded.

3d.

Audio streams from each original MP4 file were transcoded into MOV files with the same AAC codec and a MOV container using the following command:

```
ffmpeg -i filename.mov -c copy filename.mp4
```

Audio stream hash values were obtained and recorded for each transcoded MP4 file in FFmpeg 5.1 using the following command:

```
ffmpeg -i filename.mp4 -map 0:a:0 -f streamhash –
```

Comparisons between audio stream hashes for original MP4 files and MOV files derived from original MP4 files were observed and recorded.

****Each of these steps was repeated in FFmpeg 6.0****

CHAPTER IV

RESULTS

A comparison of the audio bit stream hash values of the original files was conducted within each group of original files and their respective derivatives. The entirety of the results are consistent within each group of 10 derivatives created from an original file. All comparisons were classified into at least one of the following categories:

Consistent between original files and derivatives within FFmpeg 5.1

Inconsistent between original files and derivatives within FFmpeg 6.0

Consistent between original files across FFmpeg 5.1 and FFmpeg 6.0

Inconsistent between original files across FFmpeg 5.1 and FFmpeg 6.0

Consistent between derivative files across FFmpeg 5.1 and FFmpeg 6.0

Inconsistent between derivative files across FFmpeg 5.1 and FFmpeg 6.0

Results

The complete list of original files, derivative files, their respective audio stream hash values and version of FFmpeg used to create each derivative file and calculate the corresponding audio stream hash value for each file are listed in the collection of tables below. Each table/file group is comprised an original file group (MOV, MP4, WAV, MP3) followed by all file groups derived from the original file group. Original MOV files and all of their derivatives are colored yellow. Original MP4 files and all of their derivatives are colored green. Original WAV files and all of their derivatives are colored blue. Original MP3 files and all of their derivatives are colored pink.

Table 9: Audio stream hash of original MOV video files with ACC codec using FFmpeg 5.1

File Name	SHA256 Audio Stream Hash
Balloon.MOV	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c
Door.MOV	a66185c634b948db1a844a8ec3fac3af54f7678e0d2685b09dd8ac683fc74d31
Flowers.MOV	7d36e3502569f22a7e8663d74321046670cd8da178bc1665ed3ec496b9206f9c
Rocks.MOV	a7545f62497ff82a0b34ef0602a31c032f56a9960b316d3ed69dbc3f88dc1939
Run.MOV	555347adeb201efa5254735f3023cca73862346b710de4542813d2efe144f712
Scooter.MOV	ba1af1ce5897effa4568c137ddbe9b5d11a11c006efa26950abc3b8d66a1f44c
Shadow.MOV	a0fb331cb8ecb278389bb10311675ab05b22f4ae0873726e0090d7d7bee76f8d
Sun.MOV	e4e0a1e54b004998745bc9f4b7f9243acb8bee7f075e39068ffcb31f9e1055af
Tomato.MOV	c7e08ca0b3c548913a1d28703f25963d4a73781e232393dc3a4bb977c97e1a8b
Trees.MOV	e9615405d53ab42d803bea983fced9a9211d8e65663685be500a4d8ed6108abb

Table 10: Audio stream hash of original MOV video files with ACC codec using FFmpeg 6.0

File Name	SHA256 Audio Stream Hash
Balloon.MOV	9a8d8e4c16bda02ff3c95306db968e2b72100af3c5488f99872e72353c92b9c9
Door.MOV	ea62300bb115ed1b03b39da0f4383f913e257acf55e4cdf133537a9f1981f93a
Flowers.MOV	b8872b128af1e906b7e7604e334dce05f0d00ce4e211684ea5526bf53ee3a1a7
Rocks.MOV	44c27e09c9b60317baf53fcce5779f22f93fcd4bee23cca9d17d064f74e5f690
Run.MOV	9a26188b63c7180eb6ffe52989924aa27b9b7165f6d6948d03f5cf2642f4e574
Scooter.MOV	1d8a38c370432fd537bd0b395b0bea733f1832b26183aaa1e003b56c3a7d20da
Shadow.MOV	924a42e48dfa974c17131c0d9b381f1a583e8293dc5141c6216d5afd3b0bee83
Sun.MOV	b4a86f9d5afbbb968e6e313c78364cdcff17df098c308c295ebd05559d866e7a
Tomato.MOV	c14c896e74dfecc12c6f99439c3c787c43763bad3f9d26ccc3b874e220e4297a
Trees.MOV	ae7d1fb5d9a81fddf41324cdf1a76ca116c74a72fe9e9f1999428fedeaca80b5

Table 11: Audio stream hash of WAV files derived from original MOV files in FFmpeg 5.1

File Name	SHA256 Audio Stream Hash
BalloonMOV_ff5.wav	18f6e1e52a1b22a86cc3a42e53eb12179cf43811385e1a3198dacdb1638c21c8
DoorMOV_ff5.wav	22e7699ef1b28b4bf5c62d1ce3a02c26e01ed31e668976ff14a58ae466139f4f
FlowersMOV_ff5.wav	c060b75a71b87ec5d5340e98d67d2afb201e143f7eb054c8decbbfd1c6c3e672
RocksMOV_ff5.wav	37e6ce46b1adc4ae415209db0593ff2f6447520c0289bc62940338b1816df6f8
RunMOV_ff5.wav	c4ad30074b5f65eb8f82a275ddc6db76de02976acea3705c552713626c1abbf3
ScooterMOV_ff5.wav	c811a28e7741786807e5e4e470cff548cd0eaa26b803b13485adae5e178cfd8d
ShadowMOV_ff5.wav	fc25942717aa8c0f9eda11e0520868bfa490b25249178d9d749214d108ccca54
SunMOV_ff5.wav	69b3f37d144fadf2d59493bbd29db5a9ad9402305e43bad295fa01fce50e3d6c
TomatoMOV_ff5.wav	ccfc9a7623f439e92c724648ad3e51163b428fcc4fffcb0159bfe0735708b80d
TreesMOV_ff5.wav	47c58da36a782741e67a8982482889c2d16b1f3acec3c12e8f4874aa3b9ccce5

Table 12: Audio stream hash of WAV files derived from original MOV files in FFmpeg 6.0

File Name	SHA256 Audio Stream Hash
BalloonMOV_ff6.wav	f0d4d51252713652303e37eb96e97b3676c93de48eaf530ebafb1674ed09fa88
DoorMOV_ff6.wav	5b1bb01ab37ca6ce30d94c2bcf331061e928f1d696061d6eea4a67f95ba6d604
FlowersMOV_ff6.wav	3f85c8466630c86f9c8a1855fb003a1acde01870b5b7d590371a3ac4bf0b4140
RocksMOV_ff6.wav	37e6ce46b1adc4ae415209db0593ff2f6447520c0289bc62940338b1816df6f8
RunMOV_ff6.wav	0ac1e2ed1236e10badae6e9b6ac2cd8f8819880189ab72d7875a13c9b04c1e7d
ScooterMOV_ff6.wav	c959708be811f1bb2402f21ec8a256d695cd888407e410307b65c50f303acd3d
ShadowMOV_ff6.wav	bd4505ffa83ab4b9a45bcf93a95cc609ebeb8d7b887c2975c9d3574e74fc436b
SunMOV_ff6.wav	e81483f94549fe4588999e27def85c11c8cca227988fa3f2608941ad652dcf13
TomatoMOV_ff6.wav	ccfc9a7623f439e92c724648ad3e51163b428fcc4fffcb0159bfe0735708b80d
TreesMOV_ff6.wav	d7be383b1eb7270a02724219009c0844bc2df9e3222c956be87e0fe91df449aa

Table 13: Audio stream hash of M4A files derived from original MOV files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
BalloonMOV_ff5.m4a	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c
DoorMOV_ff5.m4a	a66185c634b948db1a844a8ec3fac3af54f7678e0d2685b09dd8ac683fc74d31
FlowersMOV_ff5.m4a	7d36e3502569f22a7e8663d74321046670cd8da178bc1665ed3ec496b9206f9c
RocksMOV_ff5.m4a	a7545f62497ff82a0b34ef0602a31c032f56a9960b316d3ed69dbc3f88dc1939
RunMOV_ff5.m4a	555347adeb201efa5254735f3023cca73862346b710de4542813d2efe144f712
ScooterMOV_ff5.m4a	ba1af1ce5897effa4568c137ddbe9b5d11a11c006efa26950abc3b8d66a1f44c
ShadowMOV_ff5.m4a	a0fb331cb8ecb278389bb10311675ab05b22f4ae0873726e0090d7d7bee76f8d
SunMOV_ff5.m4a	e4e0a1e54b004998745bc9f4b7f9243acb8bee7f075e39068ffcb31f9e1055af
TomatoMOV_ff5.m4a	c7e08ca0b3c548913a1d28703f25963d4a73781e232393dc3a4bb977c97e1a8b
TreesMOV_ff5.m4a	e9615405d53ab42d803bea983fced9a9211d8e65663685be500a4d8ed6108abb

Table 14: Audio stream hash of M4A files derived from original MOV files in FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
BalloonMOV_ff6.m4a	9a8d8e4c16bda02ff3c95306db968e2b72100af3c5488f99872e72353c92b9c9
DoorMOV_ff6.m4a	ea62300bb115ed1b03b39da0f4383f913e257acf55e4cdf133537a9f1981f93a
FlowersMOV_ff6.m4a	b8872b128af1e906b7e7604e334dce05f0d00ce4e211684ea5526bf53ee3a1a7
RocksMOV_ff6.m4a	44c27e09c9b60317baf53fcce5779f22f93fcd4bee23cca9d17d064f74e5f690
RunMOV_ff6.m4a	9a26188b63c7180eb6ffe52989924aa27b9b7165f6d6948d03f5cf2642f4e574
ScooterMOV_ff6.m4a	1d8a38c370432fd537bd0b395b0bea733f1832b26183aaa1e003b56c3a7d20da
ShadowMOV_ff6.m4a	924a42e48dfa974c17131c0d9b381f1a583e8293dc5141c6216d5afd3b0bee83
SunMOV_ff6.m4a	b4a86f9d5afbbb968e6e313c78364cdcff17df098c308c295ebd05559d866e7a
TomatoMOV_ff6.m4a	c14c896e74dfecc12c6f99439c3c787c43763bad3f9d26ccc3b874e220e4297a
TreesMOV_ff6.m4a	ae7d1fb5d9a81fddf41324cdf1a76ca116c74a72fe9e9f1999428fedeaca80b5

Table 15: Audio stream hash of MP4 files derived from original MOV files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
BalloonMOV_ff5.mp4	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c
DoorMOV_ff5.mp4	a66185c634b948db1a844a8ec3fac3af54f7678e0d2685b09dd8ac683fc74d31
FlowersMOV_ff5.mp4	7d36e3502569f22a7e8663d74321046670cd8da178bc1665ed3ec496b9206f9c
RocksMOV_ff5.mp4	a7545f62497ff82a0b34ef0602a31c032f56a9960b316d3ed69dbc3f88dc1939
RunMOV_ff5.mp4	555347adeb201efa5254735f3023cca73862346b710de4542813d2efe144f712
ScooterMOV_ff5.mp4	ba1af1ce5897effa4568c137ddbe9b5d11a11c006efa26950abc3b8d66a1f44c
ShadowMOV_ff5.mp4	a0fb331cb8ecb278389bb10311675ab05b22f4ae0873726e0090d7d7bee76f8d
SunMOV_ff5.mp4	e4e0a1e54b004998745bc9f4b7f9243acb8bee7f075e39068ffcb31f9e1055af
TomatoMOV_ff5.mp4	c7e08ca0b3c548913a1d28703f25963d4a73781e232393dc3a4bb977c97e1a8b
TreesMOV_ff5.mp4	e9615405d53ab42d803bea983fced9a9211d8e65663685be500a4d8ed6108abb

Table 16: Audio stream hash of MP4 files derived from original MOV files in FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
BalloonMOV_ff6.mp4	3262f2bd192f3bdb400d3bf911afe2fab37c4065edafc3fbad35ba1cd667cd45
DoorMOV_ff6.mp4	975829ab5be36ac9b6b1209a4bba61932460b605d3ece0dc29b5aa2162c7a5be
FlowersMOV_ff6.mp4	f693873636a00e216deca81a9a1debf1c76a45dc1db9299d78a981b0aa349626
RocksMOV_ff6.mp4	f82fdd59ec92f31f14e05651c79fb557c9bee3f52c44ffd225f6c314fda6566b
RunMOV_ff6.mp4	b8135d1b8e85c1b08342b531d0c81a42eb7f2b5123e2ba57549363ee3fce709c
ScooterMOV_ff6.mp4	0f8a701a24188c068750e659061ccc9c19c5e79d2e7c400f28b752ff5a80354e
ShadowMOV_ff6.mp4	097b5a61f965a762841e117b0e066411610da511b52b0e74ca5bc4461899c35a
SunMOV_ff6.mp4	ce2a65773cfdebde8448bbe090c2a931de426a191fecbb89fecf63ba8a04e8ba
TomatoMOV_ff6.mp4	8e782d2c3510ab7093abe5bd64379bfc6160422ee9af1d3db9e9555c34ff6ac0
TreesMOV_ff6.mp4	d4aad1f8e83f953d72734853b3a213f42877d9d680a17e127829a9df89a68744

Table 17: Audio stream hash of original MP4 video files with AAC codec using FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Barry.MP4	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b
DirtBike.MP4	fd82cbe56cec031be40dff20fadc3596fa9575be1289d7da9fd01c2655806a52
Horse.MP4	7139ccb3c3e87304d52e157979b9087876a73cd55017e04a3c8e2301e96c2355
Pistons.MP4	26a62b8ed7fe84661b3234b206ed082e0388176ca8f06db03cfc0772c9ba762a
RedWings.MP4	920dd6b0eba56ce442adc1b1a796ec183d2f7140fa4cc7b6b68467a6d3945675
Rollerblades.MP4	b3ec4d66f4c13b2fd03186940a640332a895f71d62bb3dd09c7e5906b1e6d9e8
Skateboard.MP4	f55f6ec981572e67f289246d4930a271af1fad31d07a1ebd21d0a62051befac5
Surfing.MP4	aff701144f72f0e52eea3bf3333a68e73209d7f2434ea807372bb89b80aa77de
Tigers.MP4	9ea93878841b93661c79aebd19453d460b6b8e14134637f2f34dacd0dd21f686
Track.MP4	9e3b72d85f6d3e12bdc81d03f103928e08260494ec599d9f78ab54bf7f3dee68

Table 18: Audio stream hash of original MP4 video files with AAC codec using FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
Barry.MP4	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e
DirtBike.MP4	40ab05663db984b01f32fe8093f0d242a0784c1b06ee90bf44391f1b870dd21c
Horse.MP4	4fe3c669656266925ec58c9b6b5f6467dfd7a8a11caa7326f413738173ffd134
Pistons.MP4	c11dc90d18e544b03f8094ada034dba1808da7ebdf98433d9d60c4bb728e6086
RedWings.MP4	1819640b0f8a00e1ff8d7ec38d4fe625beb663b15515c182a31a6843945c4cba
Rollerblades.MP4	bf5e0627cc74245205f87cfcee04882bc521bbbb81db79aab07004ddec887c10
Skateboard.MP4	9dab3022e7210e067673dc551a03ee009d2510a989bf033018693f3f29038372
Surfing.MP4	bacd0c2e94cd698f3016e8dd1b6a1770027f4301bd700a1e878735bc480e7af2
Tigers.MP4	75e72e8a4a62fa379bcaedca7df5f55516048bf312b3db4199ff7ecbda10830f
Track.MP4	1f2693b33ec7c24bdecf57fba349c3ed2048ed5cd4ecac2695ee3773769d3dd9

Table 19: Audio stream hash of WAV files derived from original MP4 files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
BarryMP4_ff5.wav	13fbbf214aff7deb5b15bf3b4cd3e15ff3f9ae53ab7403ab53accae4ecc1d5f4
DirtBikeMP4_ff5.wav	13e7459e79e34fdfa7e4c120d2c6635f26f12bb15807079c9520f2fedb4d1965
HorseMP4_ff5.wav	8d92be1cfd544a10b2496622ede2e36b094ff0c34f0a203c68a2991a23a9d833
PistonsMP4_ff5.wav	220ad9931e6f5429d30d245566397780b1057ad18830e9360ed7c6585b2f309b
RedWingsMP4_ff5.wav	cb3b2c978c09683a5ab5c3c7fe3305488ba3c1b9bcd11275cdacd1d53effa773
RollerbladesMP4_ff5.wav	bc93bec1e3c35569174e502a550a5f4a43794a8f320d56bf8835a07890034a9c
SkateboardMP4_ff5.wav	19dd69cb7c5981a51dde66cf9f5ffdc126f48f3f6edd916de12cd6c911d31308
SurfingMP4_ff5.wav	eb215690f125505a97603ce9259bc5cd5af57e418ffd5427558ad2df47ddb862
TigersMP4_ff5.wav	866e45a1a4cf7ba3f88b52b409f3b6479e9ca7f2dec1c65df62fa0d769ebb996
TrackMP4_ff5.wav	0724fe28f0375ef313404c8959ee747b41fc9b2ff0e01d04bb8a599a02043518

Table 20: Audio stream hash of WAV files derived from original MP4 files in FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
BarryMP4_ff6.wav	eeb2ff1858b2e38695265054c04057a5180cc771ba10ce6c6eb2815689b85655
DirtBikeMP4_ff6.wav	6a92bb4c46fdf51bf27730d8780f493806c57310deb2936ef208d3b8a14cd8af
HorseMP4_ff6.wav	e23db099dbaf33077cfad580c5d1cb7744fa7425e15276017fc3bc0784b88e2f
PistonsMP4_ff6.wav	f3876ebfd03573896e1f91271dde886cd3ee52139feac35166e8625aa8da334c
RedWingsMP4_ff6.wav	9cb798fba7ebcf40dba4e1e10b64d30351560bbcb1ca9c5f9c3b9c0ce1e6cca5
RollerbladesMP4_ff6.wav	11af971c7918f3f896c825204efb28782f8b11fbd5cfbc2dc7dba24f3c2f69a0
SkateboardMP4_ff6.wav	73a62c0a50fdd06b59777406e5d34642c9f8011395878a117ca69c5078a33742
SurfingMP4_ff6.wav	4783212a12d5e3fdb95dda22d9cfde8f6947b8b1fea747454d47da8b554da549
TigersMP4_ff6.wav	9f1263e40a901820d124fa52f52c07a88301d856af92b0322d14a08148ddb4c5
TrackMP4_ff6.wav	c03716e32f4276ac0512cb622bd2994bff7fdb1ddf45ae7cb735f88f076583c1

Table 21: Audio stream hash of M4A files derived from original MP4 files in FFmpeg 5.1

File #	File Name	SHA256 Audio Stream Hash
1	BarryMP4_ff5.m4a	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b
2	DirtBikeMP4_ff5.m4a	fd82cbe56cec031be40dff20fadc3596fa9575be1289d7da9fd01c2655806a52
3	HorseMP4_ff5.m4a	7139ccb3c3e87304d52e157979b9087876a73cd55017e04a3c8e2301e96c2355
4	PistonsMP4_ff5.m4a	26a62b8ed7fe84661b3234b206ed082e0388176ca8f06db03cfc0772c9ba762a
5	RedWingsMP4_ff5.m4a	920dd6b0eba56ce442adc1b1a796ec183d2f7140fa4cc7b6b68467a6d3945675
6	RollerbladesMP4_ff5.m4a	b3ec4d66f4c13b2fd03186940a640332a895f71d62bb3dd09c7e5906b1e6d9e8
7	SkateboardMP4_ff5.m4a	f55f6ec981572e67f289246d4930a271af1fad31d07a1ebd21d0a62051befac5
8	SurfingMP4_ff5.m4a	aff701144f72f0e52eea3bf3333a68e73209d7f2434ea807372bb89b80aa77de
9	TigersMP4_ff5.m4a	9ea93878841b93661c79aebd19453d460b6b8e14134637f2f34dacd0dd21f686
10	TrackMP4_ff5.m4a	9e3b72d85f6d3e12bdc81d03f103928e08260494ec599d9f78ab54bf7f3dee68

Table 22: Audio stream hash of M4A files derived from original MP4 files in FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
BarryMP4_ff6.m4a	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e
DirtBikeMP4_ff6.m4a	40ab05663db984b01f32fe8093f0d242a0784c1b06ee90bf44391f1b870dd21c
HorseMP4_ff6.m4a	4fe3c669656266925ec58c9b6b5f6467dfd7a8a11caa7326f413738173ffd134
PistonsMP4_ff6.m4a	c11dc90d18e544b03f8094ada034dba1808da7ebdf98433d9d60c4bb728e6086
RedWingsMP4_ff6.m4a	1819640b0f8a00e1ff8d7ec38d4fe625beb663b15515c182a31a6843945c4cba
RollerbladesMP4_ff6.m4a	bf5e0627cc74245205f87cfcee04882bc521bbbb81db79aab07004ddec887c10
SkateboardMP4_ff6.m4a	9dab3022e7210e067673dc551a03ee009d2510a989bf033018693f3f29038372
SurfingMP4_ff6.m4a	bacd0c2e94cd698f3016e8dd1b6a1770027f4301bd700a1e878735bc480e7af2
TigersMP4_ff6.m4a	75e72e8a4a62fa379bcaedca7df5f55516048bf312b3db4199ff7ecbda10830f
TrackMP4_ff6.m4a	1f2693b33ec7c24bdecf57fba349c3ed2048ed5cd4ecac2695ee3773769d3dd9

Table 23: Audio stream hash of MOV files derived from original MP4 files in FFmpeg 5.1

File Name	SHA256 Audio Stream Hash
BarryMP4_ff5.MOV	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b
DirtBikeMP4_ff5.MOV	fd82cbe56cec031be40dff20fadc3596fa9575be1289d7da9fd01c2655806a52
HorseMP4_ff5.MOV	7139ccb3c3e87304d52e157979b9087876a73cd55017e04a3c8e2301e96c2355
PistonsMP4_ff5.MOV	26a62b8ed7fe84661b3234b206ed082e0388176ca8f06db03cfc0772c9ba762a
RedWingsMP4_ff5.MOV	920dd6b0eba56ce442adc1b1a796ec183d2f7140fa4cc7b6b68467a6d3945675
RollerbladesMP4_ff5.MOV	b3ec4d66f4c13b2fd03186940a640332a895f71d62bb3dd09c7e5906b1e6d9e8
SkateboardMP4_ff5.MOV	f55f6ec981572e67f289246d4930a271af1fad31d07a1ebd21d0a62051befac5
SurfingMP4_ff5.MOV	aff701144f72f0e52eea3bf3333a68e73209d7f2434ea807372bb89b80aa77de
TigersMP4_ff5.MOV	9ea93878841b93661c79aebd19453d460b6b8e14134637f2f34dacd0dd21f686
TrackMP4_ff5.MOV	9e3b72d85f6d3e12bdc81d03f103928e08260494ec599d9f78ab54bf7f3dee68

Table 24: Audio stream hash of MOV files derived from original MP4 files in FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
BarryMP4_ff6.MOV	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e
DirtBikeMP4_ff6.MOV	40ab05663db984b01f32fe8093f0d242a0784c1b06ee90bf44391f1b870dd21c
HorseMP4_ff6.MOV	4fe3c669656266925ec58c9b6b5f6467dfd7a8a11caa7326f413738173ffd134
PistonsMP4_ff6.MOV	c11dc90d18e544b03f8094ada034dba1808da7ebdf98433d9d60c4bb728e6086
RedWingsMP4_ff6.MOV	1819640b0f8a00e1ff8d7ec38d4fe625beb663b15515c182a31a6843945c4cba
RollerbladesMP4_ff6.MOV	bf5e0627cc74245205f87cfcee04882bc521bbbb81db79aab07004ddec887c10
SkateboardMP4_ff6.MOV	9dab3022e7210e067673dc551a03ee009d2510a989bf033018693f3f29038372
SurfingMP4_ff6.MOV	bacd0c2e94cd698f3016e8dd1b6a1770027f4301bd700a1e878735bc480e7af2
TigersMP4_ff6.MOV	75e72e8a4a62fa379bcaedca7df5f55516048bf312b3db4199ff7ecbda10830f
TrackMP4_ff6.MOV	1f2693b33ec7c24bdecf57fba349c3ed2048ed5cd4ecac2695ee3773769d3dd9

Table 25: Audio stream hash of original WAV audio files with PCM codec using FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Music1.wav	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music2.wav	733b1bbf6406247cc650a4f7c74e7948052f875617d112bc53fcf07ea492483a
Music3.wav	a9d59538b375aaa4ce8192c6d522c65be52bc622130947160f23da82e781ede5
Music4.wav	8a7b407ccf9c52fcab1cbe873982580d00a50f671cca09328e0420ff833ee797
Music5.wav	b220323f95a9fd38147ff0e6bdaaacc8f80354dbdf10ecf01b793326cab1e06d
Music6.wav	5899747a434a0ed87b2d1549ed910c8665a0d23e1775315a4bd420d31758ad38
Music7.wav	55fd6799cbbc5f291592eeb4380108b50a0598a2afb6288e1bd3c570b99b9dc1
Music8.wav	fbaaef8defe6ede4e9ba5a20fd32adcd4c4179a0b1e36d88dd8c3b017c3c88b7
Music9.wav	f0a70179ba82e0ea649c260e75350ea3bb3ddd8b4c4cab47bde5d853eb05c69f
Music10.wav	db621c6fbe620ac4409ff247479e9ac89c29dce6d192230d2ee84da4ccf06871

Table 26: Audio stream hash of original WAV audio files with PCM codec using FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
Music1.wav	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music2.wav	733b1bbf6406247cc650a4f7c74e7948052f875617d112bc53fcf07ea492483a
Music3.wav	a9d59538b375aaa4ce8192c6d522c65be52bc622130947160f23da82e781ede5
Music4.wav	8a7b407ccf9c52fcab1cbe873982580d00a50f671cca09328e0420ff833ee797
Music5.wav	b220323f95a9fd38147ff0e6bdaaacc8f80354dbdf10ecf01b793326cab1e06d
Music6.wav	5899747a434a0ed87b2d1549ed910c8665a0d23e1775315a4bd420d31758ad38
Music7.wav	55fd6799cbbc5f291592eeb4380108b50a0598a2afb6288e1bd3c570b99b9dc1
Music8.wav	fbaaef8defe6ede4e9ba5a20fd32adcd4c4179a0b1e36d88dd8c3b017c3c88b7
Music9.wav	f0a70179ba82e0ea649c260e75350ea3bb3ddd8b4c4cab47bde5d853eb05c69f
Music10.wav	db621c6fbe620ac4409ff247479e9ac89c29dce6d192230d2ee84da4ccf06871

Table 27: Audio stream hash of MP3 files derived from original WAV files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Music1wav_ff5.mp3	8cab8dd5cdc2f4ec95fb43938d1da186372a26f4929dd833a0073ad4ce13c6fe
Music2wav_ff5.mp3	0410842024a09d534ae391b81f9fdceb61409d09cea5e12baa1f312feef66d94
Music3wav_ff5.mp3	f8508e3e5121dabc67c1ceb0112762712c8bd7a4daf670285a48b36fdd832759
Music4wav_ff5.mp3	047e0f374347169f40a0cdb9e884d8d63d1217277a618b8bcc626de4a220b508
Music5wav_ff5.mp3	362b98ab6025e675668fd7037a693d8a42a4f2f646421a1aeb297deb65ab9479
Music6wav_ff5.mp3	297f833392aea8b807f41d0a0336be9ef323393099a38e15d543fe28e572c308
Music7wav_ff5.mp3	9d512728463edb575c8aa9e260b67e00065fd205ac1ec975dc841686cc2f1ac5
Music8wav_ff5.mp3	36d09604d843c705f820a838df31f00dad8b7aa888201d1f142dce1970dc2991
Music9wav_ff5.mp3	b57ca97c915bdcb50984f8ba0f5d36e20a70fea44c1483a66113f97d47c886da
Music10wav_ff5.mp3	f7d69e8b06ad2f45904a2fc460fa18f8b476ed3d49b068d60835d210d1332517

Table 28: Audio stream hash of MP3 files derived from original WAV files in FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
Music1wav_ff6.mp3	8cab8dd5cdc2f4ec95fb43938d1da186372a26f4929dd833a0073ad4ce13c6fe
Music2wav_ff6.mp3	0410842024a09d534ae391b81f9fdceb61409d09cea5e12baa1f312feef66d94
Music3wav_ff6.mp3	f8508e3e5121dabc67c1ceb0112762712c8bd7a4daf670285a48b36fdd832759
Music4wav_ff6.mp3	047e0f374347169f40a0cdb9e884d8d63d1217277a618b8bcc626de4a220b508
Music5wav_ff6.mp3	362b98ab6025e675668fd7037a693d8a42a4f2f646421a1aeb297deb65ab9479
Music6wav_ff6.mp3	297f833392aea8b807f41d0a0336be9ef323393099a38e15d543fe28e572c308
Music7wav_ff6.mp3	9d512728463edb575c8aa9e260b67e00065fd205ac1ec975dc841686cc2f1ac5
Music8wav_ff6.mp3	36d09604d843c705f820a838df31f00dad8b7aa888201d1f142dce1970dc2991
Music9wav_ff6.mp3	b57ca97c915bdcb50984f8ba0f5d36e20a70fea44c1483a66113f97d47c886da
Music10wav_ff6.mp3	f7d69e8b06ad2f45904a2fc460fa18f8b476ed3d49b068d60835d210d1332517

Table 29: Audio stream hash of M4A files derived from original WAV files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Music1wav_ff5.m4a	ea036146a91aa7cd1ea28872cbf5ecffe4fe8136ac85d26e920dd66c630a111f
Music2wav_ff5.m4a	732c61f807912fb48aa6860e5965b7669772712b721517cc68581aa192ed3bc8
Music3wav_ff5.m4a	0784b41175f6f902c32e036c3b66466de894ad83577a7c2fba7ccb6b8b814886
Music4wav_ff5.m4a	e2cbfe1fb84fa18a6d99b5ad3acb13478c25e5d936785cc8f90faf5a3bdf13ef
Music5wav_ff5.m4a	c007c161dbf660838d3804076f28175c47437346dd729cec2d9244e920942ce9
Music6wav_ff5.m4a	1863b622e444d3e557abd05da969bb7dda93fa770a47b19c3e47d282fe1f0150
Music7wav_ff5.m4a	9f66ac07f8fb5621159c4942184a159473d0c6bc3106bb6294e1f956a3f5a266
Music8wav_ff5.m4a	e70b0cdba6ab723e41262bd10edc5f4f9fdc1bedfda5a41eaf23dc74a41925e4
Music9wav_ff5.m4a	ddceb73a88e8efbcc20bd6fb7f542e36f0e9afb241c8411557a09a462b7c19c7
Music10wav_ff5.m4a	699a112675cba968579d7ad2452638c2d085440e6226a3619b4e193dfb477559

Table 30: Audio stream hash of M4A files derived from original WAV files in FFmpeg 6.0

File Name	SHA256 Audio Stream Hash
Music1wav_ff6.m4a	35a2feeec7306facbdf409b35fc1baf3496e6aea1ad16368147d53d03cc70822
Music2wav_ff6.m4a	b4d7cb0aac9299999f90a4afe83b057d83fc4884b21cfc9d9c4a7f6dbf2ec92c
Music3wav_ff6.m4a	04005009834c824c99867db903dd7b966527da535cfe20aa44ce04e434ff34e6
Music4wav_ff6.m4a	d96e974fab5fd4d4723a45f403d971eee8128e4f938d5919b7f0a6fc030e4860
Music5wav_ff6.m4a	3766c14f521d9417a1b21718bfd8545d9828526aac6f6d960956455682f8f197
Music6wav_ff6.m4a	923d53783fb080fae4a3c94f3810991826eb7e7dce05cff8362972d43db80854
Music7wav_ff6.m4a	54d2695f79ecc5f2741fcfb310cf1c9dbd48ceb7dcf74ff629236453b00bbc5a
Music8wav_ff6.m4a	5053a087b2dcdf6bcbc500e0a3396346790203a80aa2bce4d1ee77b9d99ed449
Music9wav_ff6.m4a	a09d12eff3d87d02935376caf71cbff52961d9f302d5e3d2b9779afea3b58bbf
Music10wav_ff6.m4a	9222b7cfd441400ce2a26467cfa4b3b4d7d526fd932c6f1ed68ead31127121ab

Table 31: Audio stream hash of AVI files derived from original WAV files in FFmpeg 5.1

File Name	SHA256 Audio Stream Hash
Music1wav_ff5.avi	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music2wav_ff5.avi	733b1bbf6406247cc650a4f7c74e7948052f875617d112bc53fcf07ea492483a
Music3wav_ff5.avi	a9d59538b375aaa4ce8192c6d522c65be52bc622130947160f23da82e781ede5
Music4wav_ff5.avi	8a7b407ccf9c52fcab1cbe873982580d00a50f671cca09328e0420ff833ee797
Music5wav_ff5.avi	b220323f95a9fd38147ff0e6bdaaacc8f80354dbdf10ecf01b793326cab1e06d
Music6wav_ff5.avi	5899747a434a0ed87b2d1549ed910c8665a0d23e1775315a4bd420d31758ad38
Music7wav_ff5.avi	55fd6799cbbc5f291592eeb4380108b50a0598a2afb6288e1bd3c570b99b9dc1
Music8wav_ff5.avi	fbaaef8defe6ede4e9ba5a20fd32adcd4c4179a0b1e36d88dd8c3b017c3c88b7
Music9wav_ff5.avi	f0a70179ba82e0ea649c260e75350ea3bb3ddd8b4c4cab47bde5d853eb05c69f
Music10wav_ff5.avi	db621c6fbe620ac4409ff247479e9ac89c29dce6d192230d2ee84da4ccf06871

Table 32: Audio stream hash of AVI files derived from original WAV files in FFmpeg 6.0

File Name	SHA256 Audio Stream Hash
Music1wav_ff6.avi	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music2wav_ff6.avi	733b1bbf6406247cc650a4f7c74e7948052f875617d112bc53fcf07ea492483a
Music3wav_ff6.avi	a9d59538b375aaa4ce8192c6d522c65be52bc622130947160f23da82e781ede5
Music4wav_ff6.avi	8a7b407ccf9c52fcab1cbe873982580d00a50f671cca09328e0420ff833ee797
Music5wav_ff6.avi	b220323f95a9fd38147ff0e6bdaaacc8f80354dbdf10ecf01b793326cab1e06d
Music6wav_ff6.avi	5899747a434a0ed87b2d1549ed910c8665a0d23e1775315a4bd420d31758ad38
Music7wav_ff6.avi	55fd6799cbbc5f291592eeb4380108b50a0598a2afb6288e1bd3c570b99b9dc1
Music8wav_ff6.avi	fbaaef8defe6ede4e9ba5a20fd32adcd4c4179a0b1e36d88dd8c3b017c3c88b7
Music9wav_ff6.avi	f0a70179ba82e0ea649c260e75350ea3bb3ddd8b4c4cab47bde5d853eb05c69f
Music10wav_ff6.avi	db621c6fbe620ac4409ff247479e9ac89c29dce6d192230d2ee84da4ccf06871

Table 33: Audio stream hash of original MP3 audio files using FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Song1.mp3	7d5afc1b89def707229257569bde1695ea88e4d4c443bd1be16ca737d8c56d73
Song2.mp3	35e55d35066dd3542b548fb1432a22bd9f5b776ce2e6c59d161e2af34ed97373
Song3.mp3	c69e3c4559542c51857ddf57b8889da2c08e0bc2fe9847f3f398f428050900db
Song4.mp3	228eb4cb645054a48c3e89f3664fd475c347c4986f20565b95038a26e341b41f
Song5.mp3	fdb5c2e336e71e3a8f3e68537887dd1379162726f74ba3e6414b612405d45717
Song6.mp3	2ce215c012108a67b906664991d1e99e4595974b4c73ba0d2ec00f658b8ee4ce
Song7.mp3	31c47230d0a34923095878be60db71585c80b48a65017709ca619e18b7d8040c
Song8.mp3	cfc74677b220875c8e0926de1852e2507f9f140f70b0b62c7f19a5be46866758
Song9.mp3	8c5f50c843c2e19e57c17a73a8892e28f4fdd342ac1a96b84b82d5936d0f83cd
Song10.mp3	da8269766fa6a5cfe44d5578a6b638c67e79de7723831e141d093a057b7838e0

Table 34: Audio stream hash of original MP3 audio files using FFmpeg 6.0

File Name	**SHA256 Audio Stream Hash**
Song1.mp3	7d5afc1b89def707229257569bde1695ea88e4d4c443bd1be16ca737d8c56d73
Song2.mp3	35e55d35066dd3542b548fb1432a22bd9f5b776ce2e6c59d161e2af34ed97373
Song3.mp3	c69e3c4559542c51857ddf57b8889da2c08e0bc2fe9847f3f398f428050900db
Song4.mp3	228eb4cb645054a48c3e89f3664fd475c347c4986f20565b95038a26e341b41f
Song5.mp3	fdb5c2e336e71e3a8f3e68537887dd1379162726f74ba3e6414b612405d45717
Song6.mp3	2ce215c012108a67b906664991d1e99e4595974b4c73ba0d2ec00f658b8ee4ce
Song7.mp3	31c47230d0a34923095878be60db71585c80b48a65017709ca619e18b7d8040c
Song8.mp3	cfc74677b220875c8e0926de1852e2507f9f140f70b0b62c7f19a5be46866758
Song9.mp3	8c5f50c843c2e19e57c17a73a8892e28f4fdd342ac1a96b84b82d5936d0f83cd
Song10.mp3	da8269766fa6a5cfe44d5578a6b638c67e79de7723831e141d093a057b7838e0

Table 35: Audio stream hash of AVI files derived from original MP3 files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Song1mp3_ff5.avi	8e4a960572b26295613a3ad6446267293923f7544f762f413f27429ae5c2b3ee
Song2mp3_ff5.avi	145b2b02ce04a329101381301110c9d2ee9edf689edad6f852beead1e9ee9513
Song3mp3_ff5.avi	311c37619cbc5547305a2cec121af77c1ae6ecb51a77c4c21b0b6e1389c33b16
Song4mp3_ff5.avi	1021916abcfacdbf2b7c0e2866d6212b5b9935d232664768734a9de070d9999a
Song5mp3_ff5.avi	ca6d6425cb60da3369a76a2a99aa8128742693464ee7623ef05dbedca61d9bb6
Song6mp3_ff5.avi	b6d60b965ccc2eb12a81a22fb461db4ee34ec5991df9f95820c77a7e1cbb9532
Song7mp3_ff5.avi	24e697174da04e6194c935ee50451c022ccecf7ee20a92ab5fd78f4143f3ee94
Song8mp3_ff5.avi	ea6328f760b33b0db6ad9ad23cfd0baf0e4b0ec0f3d52db40eb24a85c5953f98
Song9mp3_ff5.avi	8541d584bf37f3259438f38d9478a57c6d8c8b00e5da23811d34f54976736d87
Song10mp3_ff5.avi	a69247b44bd7fb2a3913f1305106633814b088704d40294aaa39f76104357444

Table 36: Audio stream hash of AVI files derived from original MP3 files in FFmpeg 5.1

File Name	**SHA256 Audio Stream Hash**
Song1mp3_ff6.avi	8e4a960572b26295613a3ad6446267293923f7544f762f413f27429ae5c2b3ee
Song2mp3_ff6.avi	145b2b02ce04a329101381301110c9d2ee9edf689edad6f852beead1e9ee9513
Song3mp3_ff6.avi	311c37619cbc5547305a2cec121af77c1ae6ecb51a77c4c21b0b6e1389c33b16
Song4mp3_ff6.avi	1021916abcfacdbf2b7c0e2866d6212b5b9935d232664768734a9de070d9999a
Song5mp3_ff6.avi	ca6d6425cb60da3369a76a2a99aa8128742693464ee7623ef05dbedca61d9bb6
Song6mp3_ff6.avi	b6d60b965ccc2eb12a81a22fb461db4ee34ec5991df9f95820c77a7e1cbb9532
Song7mp3_ff6.avi	24e697174da04e6194c935ee50451c022ccecf7ee20a92ab5fd78f4143f3ee94
Song8mp3_ff6.avi	ea6328f760b33b0db6ad9ad23cfd0baf0e4b0ec0f3d52db40eb24a85c5953f98
Song9mp3_ff6.avi	8541d584bf37f3259438f38d9478a57c6d8c8b00e5da23811d34f54976736d87
Song10mp3_ff6.avi	a69247b44bd7fb2a3913f1305106633814b088704d40294aaa39f76104357444

Given that consistent results exist within each derivative group of transcoded files when compared to the original files, I have chosen a single derivative file from each derivative group of 10 files to represent its entire group and be used in the following data comparison. The files containing the same audio stream hash values are highlighted in the same color. Files that have an audio stream hash value that does not match the audio stream hash of any other file in the data set being compared are not highlighted in any color. This narrow focus will serve to illustrate the observed relationship between audio stream hash values of original files and their derivative files when calculated in both FFmpeg 5.1 and FFmpeg 6.0.

FILE GROUP 1: RESULTS SUMMARY

ORIGINAL MOV FILES AND DERIVATIVES

- The same original MOV files yield different audio stream hash values when they are calculated in FFmpeg 5.1 and FFmpeg 6.0 respectively
- Original MOV files yield different audio stream hash values than transcoded PCM WAV files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)
- Original MOV files yield the same audio stream hash values when compared to transcoded AAC codec M4A files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)

ORIGINAL MP4 FILES AND DERIVATIVES

- The same original MP4 files yield different audio stream hash values when they are calculated in FFmpeg 5.1 and FFmpeg 6.0 respectively
- Original MP4 files yield different audio stream hash values than transcoded PCM WAV files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)
- Original MP4 files yield the same audio stream hash values when compared to transcoded AAC codec M4A files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)

Table 37: Stream hashes for group 1 original files and derivatives – FFmpeg 5.1

FILE NAME	SHA256 AUDIO STREAM HASH
Balloon.MOV	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c
BalloonMOV_ff5.wav	18f6e1e52a1b22a86cc3a42e53eb12179cf43811385e1a3198dacdb1638c21c8
BalloonMOV_ff5.m4a	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c
Barry.MP4	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b
BarryMP4_ff5.wav	13fbbf214aff7deb5b15bf3b4cd3e15ff3f9ae53ab7403ab53accae4ecc1d5f4
BarryMP4_ff5.m4a	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b

Table 38: Stream hashes for group 1 original files and derivatives – FFmpeg 6.0

FILE NAME	SHA256 AUDIO STREAM HASH
Balloon.MOV	9a8d8e4c16bda02ff3c95306db968e2b72100af3c5488f99872e72353c92b9c9
BalloonMOV_ff6.wav	f0d4d51252713652303e37eb96e97b3676c93de48eaf530ebafb1674ed09fa88
BalloonMOV_ff6.m4a	9a8d8e4c16bda02ff3c95306db968e2b72100af3c5488f99872e72353c92b9c9
Barry.MP4	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e
BarryMP4_ff6.wav	eeb2ff1858b2e38695265054c04057a5180cc771ba10ce6c6eb2815689b85655
BarryMP4_ff6.m4a	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e

FILE GROUP 2: RESULTS SUMMARY

ORIGINAL WAV FILES

- The same original WAV files yield the same audio stream hash values when they are calculated in FFmpeg 5.1 and FFmpeg 6.0 respectively

MP3 DERIVED FROM ORIGINAL WAV FILES

- Original WAV files yield different audio stream hash values than transcoded MP3 (codec) MP3 (container) files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)
- The audio stream hashes of the same MP3 files (converted from original WAV files) are the same when calculated in FFmpeg 5.1 and FFmpeg 6.0 and compared

M4A FILES DERIVED FROM ORIGINAL WAV FILES

- Original WAV files yield different audio stream hash values than transcoded AAC (codec) M4A (container) files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)
- The audio stream hashes of M4A files (converted from original WAV files) are different when calculated in FFmpeg 5.1 than when calculated in FFmpeg 6.0 and compare

Table 39: Stream hashes for group 2 original files and derivatives – FFmpeg 5.1

FILE NAME	SHA256 AUDIO STREAM HASH
Music1.wav	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music1wav_ff5.mp3	8cab8dd5cdc2f4ec95fb43938d1da186372a26f4929dd833a0073ad4ce13c6fe
Music1wav_ff5.m4a	036146a91aa7cd1ea28872cbf5ecffe4fe8136ac85d26e920dd66c630a111f

Table 40: Stream hashes for group 2 original files and derivatives – FFmpeg 6.0

FILE NAME	SHA256 AUDIO STREAM HASH
Music1.wav	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music1wav_ff6.mp3	8cab8dd5cdc2f4ec95fb43938d1da186372a26f4929dd833a0073ad4ce13c6fe
Music1wav_ff6.m4a	35a2feeec7306facbdf409b35fc1baf3496e6aea1ad16368147d53d03cc70822

FILE GROUP 3: RESULTS SUMMARY

MOV/MP4

- The audio stream hash values for original MOV files are the same as the audio stream hash values for MP4 derivatives (with the same codec) of original MOV files (using FFmpeg 5.1 and FFmpeg 6.0)
- The audio stream hash values for original MP4 files are the same as the audio stream hash values for MOV derivatives (with the same codec) of original MP4 files (using FFmpeg 5.1 and FFmpeg 6.0)

- The audio stream hash values calculated from the same files, although matching within a single FFmpeg version, are different across versions (when the FFmpeg version 5.1 stream hash value is compared with the FFmpeg version 6.0 stream hash value)

MP3

- Original MP3 files yield different audio stream hash values than transcoded MP3 (codec) AVI (container) files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)
- The audio stream hash values for original MP3 files, when calculated in FFmpeg 5.1, are the same as the audio stream hash values for the same MP3 files, when calculated in FFmpeg 6.0

WAV

- Original WAV files yield the same audio stream hash values as transcoded PCM (codec) AVI (container) files derived from the originals (valid for FFmpeg 5.1 and FFmpeg 6.0)
- The audio stream hash values for original WAV files, when calculated in FFmpeg 5.1, are the same as the audio stream hash values for the same WAV files, when calculated in FFmpeg 6.0

Table 41: Stream hashes for group 3 original files (MP4//MOV) and derivatives – FFmpeg 5.1

FILE NAME	SHA256 AUDIO STREAM HASH
Barry.MP4	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b
BarryMP4_ff5.MOV	8c4fe6b8ba4a5c191bf27741fc2b40293c12188d6c1435ce3488f92a04b1f38b
Balloon.MOV	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c
BalloonMOV_ff5.MP4	08d2b126ff4022f9f90be2982a9c40cfefc51671572d8ff9f6326ed26cfa7e0c

Table 42: Stream hashes for group 3 original files (MP4/MOV) and derivatives – FFmpeg 6.0

FILE NAME	SHA256 AUDIO STREAM HASH
Barry.MP4	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e
BarryMP4.MOV	45b9b08e455bf8355c5f898b54c3b3e2aa10344858baad1c92fc40a38272bd7e
Balloon.MOV	3262f2bd192f3bdb400d3bf911afe2fab37c4065edafc3fbad35ba1cd667cd45
BalloonMOV_ff6.MP4	3262f2bd192f3bdb400d3bf911afe2fab37c4065edafc3fbad35ba1cd667cd45

Table 43: Stream hashes for group 3 original files (MP3/WAV) and derivatives – FFmpeg 5.1

FILE NAME	SHA256 AUDIO STREAM HASH
Song1.mp3	7d5afc1b89def707229257569bde1695ea88e4d4c443bd1be16ca737d8c56d73
Song1mp3_ff5.avi	8e4a960572b26295613a3ad6446267293923f7544f762f413f27429ae5c2b3ee
Music1.wav	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music1wav_ff5.avi	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32

Table 44: Stream hashes for group 3 original files (MP3/WAV) and derivatives – FFmpeg 6.0

FILE NAME	SHA256 AUDIO STREAM HASH
Song1.mp3	7d5afc1b89def707229257569bde1695ea88e4d4c443bd1be16ca737d8c56d73
Song1mp3_ff6.avi	8e4a960572b26295613a3ad6446267293923f7544f762f413f27429ae5c2b3ee
Music1.wav	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32
Music1wav_ff6.avi	45b90ed4a09bd5088877bba531a6a8ed4f4358355c65b826710b303d540bec32

In summary, FFmpeg functioned in ways that displayed internal consistency within its respective versions. There is no indication that transcoding files within a single version of FFmpeg directly impacted the ability of the software to properly authenticate bit stream hashes or detect differences. AAC bit streams remained unchanged and produced the same stream hash values when transcoded into different file container types in either FFmpeg 5.1 or FFmpeg 6.0, which is not surprising. However, the audio stream hash values calculated in FFmpeg 5.1 for AAC encoded data streams were different than the audio stream hash values calculated in FFmpeg 6.0 for the same files.

CHAPTER V

CONCLUSIONS

The stream hashing function for audio streams in FFmpeg appears to have functioned as expected when applied to files containing encoded audio stream data in PCM and MP3 codecs, regardless of the associated file container type. This was noted when comparing audio stream hash values for original files with their transcoded derivative files where the codec was preserved. Original and derivative files containing MP3 and PCM codecs also yielded the same audio stream hash values when compared across versions 5.1 and 6.0 of FFmpeg. The stream hashing function in FFmpeg also demonstrated the ability to detect differences in audio data when the codec for audio data was altered, thus, changing the compression.

Although audio streams encoded in AAC yielded hash values that were consistent within a single version of FFmpeg (5.1 or 6.0), the same audio streams were found to conflicting hash values when calculated using the different versions of FFmpeg. In fact, any file (original or derivative) containing AAC encoded audio data was shown to have different audio stream hash values when compared across FFmpeg versions. This is problematic because it casts doubt on the reliability of FFmpeg to produce accurate stream hashing results for AAC encoded audio data. As a result, several possibilities exist and important questions must be presented for further research.

1. Why is the audio stream hash value consistently different across different FFmpeg versions (5.1 and 6.0) for the same audio streams of data encoded in AAC codec? Which FFmpeg version (if any) is applying a correct stream hashing process/algorithm?

2. It is possible that the AAC codec is somehow encoding/decoding the data differently in FFmpeg 5.1 than it is in FFmpeg 6.0 during the stream hashing process, resulting in different hash values for the audio data stream?
3. It is possible that the stream hashing algorithm being applied during the stream hashing process in FFmpeg 5.1 is different than the stream hashing algorithm being applied in FFmpeg 6.0.
4. The transcoding process does not appear to influence the discrepancies observed in this study. Even original files that had not been subjected to any transcoding exhibited one audio stream hash value in FFmpeg 5.1 and a different audio stream has value in FFmpeg 6.0.
5. Is it possible that the discrepancy in stream hashes across different versions of FFmpeg also exists among image and video bit streams?

Future Research

Inconsistencies related to the audio stream hashing function in FFmpeg 5.1 and FFmpeg 6.0 were uncovered during this study. Different stream hash values were calculated for the same audio data streams, depending on the version of FFmpeg being used – 5.1 or 6.0. This occurred despite an original file's audio stream having the same hash value as a derivative file's audio stream within the same version of FFmpeg - 5.1 or 6.0. Moreover, this occurred regardless of the file's status as an original or a derivative.

Future research should be conducted to explore the AAC encoders used in FFmpeg version 5.1 and FFmpeg version 6.0 on a deeper level. Earlier versions of FFmpeg as well as future versions should be subjected to an analysis of stream hashing consistency to establish confidence in results. Finally, more file container types and codecs should also be included in future experiments. While this study extended beyond the scope of the initial research but there are many more file container types and codecs that are available for testing.

REFERENCES

1. Audacity. 2023. https://support.audacityteam.org/basics/installing-ffmpeg. Accessed 10 Oct 2023.

2. FFmpeg. 2023. FFmpeg codecs documentation. https://ffmpeg.org/ffmpeg-codecs.html#Audio-Encoders. Accessed 10 Oct 2023.

3. NIST. 2023. NIST policy on hash functions. https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions. Accessed 10 Oct 2023.

4. Scientific Working Group on Digital Evidence (SWGDE). SWGDE Technical notes on FFmpeg for forensic audio examination. Vol. V 1.0 2023. https://drive.google.com/file/d/1ooQNUQp-EG79nfGUokBu5VsDvyhjDWnf/view. Accessed 05 Oct 2023.

5. Scientific Working Group on Digital and Multimedia Evidence (SWGDE). SWGDE best practices for forensic audio. Vol. V 2.5. 2022. https://drive.google.com/file/d/1zF0KzO5Xe5jAO2CgLxISIm8fLSFJ46ax/view. Accessed 05 Oct 2023.

6. Wales, G. (2019, May 18). *National Center for Media Forensics Research*. https://artsandmedia.ucdenver.edu/areas-of-study/national-center-for-media-forensics/national-center-for-media-forensics-research

7. Wales, GS, Smith, JM, Lacey, DS, Grigoras, C. Multimedia stream hashing: A forensic method for content verification. *J Forensic Sci*. 2023; 68: 289–300. https://doi.org/10.1111/1556-4029.15148

APPENDIX

FFmpeg Commands Used

1. **ffmpeg -i filename.ext -map 0:a:0 -f streamhash –**

Calculate the SHA 256 stream hash of the first audio stream of the first input file

2. **ffmpeg -i filename.ext -vn -acodec pcm_s16le -ar 44100 -ac 1 filename.wav**

Bifurcate audio from original file (ignore video) and transcode it to a PCM codec, WAV container

3. **ffmpeg -i filename.ext -vn -acodec copy filename.ext**

Copy audio codec from input file (ignore video) to newly created output file

4. **ffmpeg -i filename.ext -copy c filename.ext**

Copy codec from input file to newly created output file

5. **ffmpeg -version**

Display FFmpeg version information