

AUTHENTICATING IOS 12, 13, AND 14 VIDEOS
USING METADATA AND HEX CODE

by

ANDREW ELLIS WEST

B.S, Utah Valley University, 2018

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Master of Science
Recording Arts Program

2020

© 2020

ANDREW ELLIS WEST

ALL RIGHTS RESERVED

This thesis for the Master of Science degree by

Andrew Ellis West

has been approved for the

Recording Arts Program

by

Catalin Grigoras, Chair

Cole M. Whitecotton

Gregory S. Wales

Date: December 12, 2020

West, Andrew Ellis (M.S., Recording Arts Program)

Authenticating iOS 12, 13, and 14 Videos Using Metadata and Hex Code

Thesis directed by Associate Professor Catalin Grigoras

ABSTRACT

The ability to record, edit, and share videos using the same device has created the culture behind the phrase, “pics or it didn’t happen”. This new shift in culture has allowed consumers to record and share as much as they can with their friends and family. Leading the way for this way of communicating is the popular Apple smartphone line. Its popularity stems from its impressive line of smartphone and tablet products, making iOS one of the main operating systems for consumer technology. The prevalence of this lifestyle change has also increased the number of videos taken by bystanders or suspects while crimes are taking place. With the ability to natively edit videos introduced in iOS 12, this causes concern for forensic analysts who have to authenticate these videos. In order to prove that we know a video has been edited using the native “Photo” application, we must create an approach on how to authenticate the video metadata found within the file.

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

Approved: Catalin Grigoras

DEDICATION

I would like to dedicate this paper and research to my loving wife, Miranda. Without her loving support, I wouldn't have been able to accomplish this. I would also like to dedicate this to my wonderful parents, who from the beginning have supported my crazy decision to pursue a career in digital media. I would also like to dedicate this to the many other family members and friends that have supported me all these years.

TABLE OF CONTENTS

CHAPTER

I.	INTRODUCTION	1
	Apple Device History and Relevancy	1
	Camera Culture	1
	What Makes Camera Culture Possible?	2
	iOS Versions	2
	How This Applies to Media Forensics.....	3
II.	RESEARCH AND VERIFICATION	4
	Recording a Video on an iDevice	4
	iDevice Camera Settings	6
	Photos Application Editing Capability	8
	Comparing Metadata Reader Tools	12
III.	METHODOLOGY	14
	Verifying Transfer Methods.....	14
	AirDrop and Image Capture Verification	15
	iCloud Verification	16
	Video Files from Various Past OS.....	17
IV.	METHOD	19
	List of Equipment and Resources	19
	Process for Trimming the Video	19
	Process for Cropping the Video	20
V.	RESULTS	21

Reading Metadata	21
Changes Found in the Metadata	21
Small Factors and Observations	21
Necessary Metadata.....	21
Clean Aperture Width and Height.....	22
Frame Resolution and Frame Rate	24
Encoded and Tagged Date.....	24
Source Duration.....	25
GOP and Reference Frames	26
Source Stream Duration	27
Hoov Movie Data Header	27
Decision Tree	29
VI. CONCLUSIONS	31
Possible Future Topics of Research	32
REFERENCES	33
APPENDIX.....	35
Apple Device iOS Compatibility, Camera Resolution, and Frame Rate	35

CHAPTER I

INTRODUCTION

Apple Device History and Relevancy

The first iPhone was released on June 29th, 2007 and since then, it is estimated that 1.46 billion iPhones have been sold since that day [1]. The iPad was first released on April 3rd, 2010. It's estimated that 360 million iPads have been sold since that day [2]. Since then, these two devices have permeated the smart phone and tablet market.

In 2017, there were 120 million iPhones in use in the USA and more throughout the world. In 2019, Apple held 19% of the global smartphone market share in the fourth quarter [3]. With a vast number of smartphones available to consumers, almost 1/5 of the world's smartphone users choose to use an iPhone. This means that the iPhone is a relevant tool to the world.

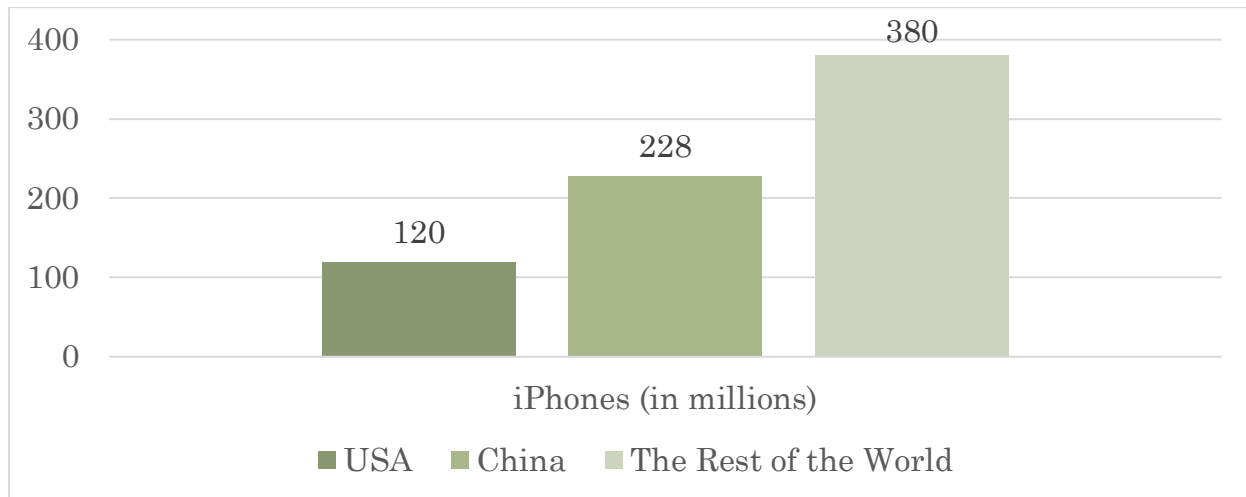


Figure 1. iPhones in Use Worldwide in 2017 [3]

Camera Culture

At the time of this thesis' publication, current social and entertainment trends promote creative content from consumers. From high resolution videos found on social media, travel

vlogs on video streaming services, and more, consumers demand quality content from their devices. The solution for high quality video recording has become available through IOS devices. We have seen this increase with Apple's push for better camera(s) on their iPhone product line. The iPhone 11 was marketed having the ability to film a professional video just using this device. An example of this was Apple's iPhone 11 commercial titled, "Snowbrawl" in which an iPhone 11 was the only camera used to film a high-resolution and high-quality commercial demonstrating the phone's video recording capabilities [4].

What Makes Camera Culture Possible?

Three variables make this camera culture possible with Apple's devices.

1. The camera(s): these amazing cameras allow the individual to take high quality videos. Each device may have a differing number of cameras or recording capability. For example, the iPhone 12 Pro has three cameras while older iPhones have 1 or 2 cameras.
2. The price entry point: at the time of this thesis publication, consumers are able to purchase an iPhone with a single high-quality camera or more for a range between \$399 and \$1,399. This wide range of prices allow users of all economical positions purchase a new iOS device with used devices being even less expensive.
3. The Operating System (OS): the OS allows these high definition videos to be recorded, processed, and shared while keeping the quality as high as possible.

iOS Versions

The first iOS to allow video editing using the native Photo application was iOS 12. Released on September 17, 2018, the operating system introduced the ability to trim your videos on the native Photos application [5]. Is still in use for phones that do not support iOS 13. List of compatible devices can be found in Appendix A.

Apple next operating system, iOS 13 was released on September 19, 2019. This operating system introduced the ability to crop your videos on the native Photos application. Users could also save their videos as new files instead of saving over the original file. Apple also announced that iPad's would no longer use iOS. Instead, the operating system would be replaced by iPadOS starting with iPadOS 13. List of iOS 13 and iPadOS 13 compatible devices can be found on Appendix A.

The most recent iOS has been iOS 14 and iPadOS14 which released on September 16, 2020 to all phones that supported iOS 13 [7]. This new iOS did not introduce any new features to the Photo application's video editing capability. A list of iOS 14 and iPadOS 14 compatible devices can be found on Appendix A.

How This Applies to Media Forensics

The ability to record high quality videos in a portable device has allowed individuals to quickly record anything at any time. We have seen an increase of cell phone recorded videos showing a wide variety of situations from civil unrest to protests. If these recorded videos on consumers phones are included in court cases, they need to be authenticated. In iOS 12, 13, and 14's native Photo application, you can edit the video by trimming, cropping, saving over the video, or saving as a new video. We are not aware of any research on how iOS 12, 13, and 14 videos may differ if altered in the native Photos application. With how prevalent iOS devices are today, it is necessary to create an approach to authenticating iOS 12, 13, and 14 video files. What happens though, if a video is edited before being admitted as evidence? How can we authenticate that the file submitted as evidence has not been altered in the Photo application?

CHAPTER II

RESEARCH AND VERIFICATION

Throughout this thesis, I will be referring to devices that operate on iOS 12, 13, and 14 as “iDevices”. This includes the iPhone, iPad, and iPod Touch product line. The research conducted will be on the iDevice’s camera and its recording options and does not include the audio found within the video. It is outside the scope of this thesis but may be investigated in a future thesis.

Recording a Video on an iDevice

Recording videos on an iDevice can be done by going to the camera application. Within there, you can select several different options. Figure 2 demonstrates how the Camera application looks and operates on the iPhone XS Max on iOS 14.0. Other iDevices user interface for their camera application may appear differently however, they all do the same thing.



Figure 2. Camera Video Recording User Interface

1. Camera Flash: You can choose to turn the flash on, off, or automatically.
2. Timecode: This is how long you have been recording for since hitting the red record button. Only found in the video recording section.
3. Resolution and Frame Rate: The resolution and frame rate you will film in are displayed here. You can click on either the resolution or frame rate to change it.
4. Camera Zoom: This shows you how far zoomed in you are. Some devices with multiple camera lenses will switch between which one is in use depending on the zoom level. You can easily switch between the lenses by clicking on this button.

5. Photos/Videos Quick View: You can view what images and videos you've taken recently.
6. Record: Pressing this button will start recording video. Pressing the button again will stop the recording. The recorded video will be saved to the Photo application. There is no pause and resume recording function.
7. Front/Back Camera: Switch between the front and back camera.

For extensive guides on how to use the camera on any iDevice, visit Apple Support at support.apple.com.

iDevice Camera Settings

In order to understand what information can be changed, we need to look at what camera settings each device has. What settings in the Camera application can we change, and will it have an impact on the metadata? These settings can be found in *Settings -> Camera -> Formats*.

- **Formats:** iDevices can film in two different video codecs. Both codecs create a .mov file.
 - **High Efficiency:** Uses the High Efficiency Video Codec (HEVC) which is also known as MPEG-H or H.265. This codec can be used for all videos but is required when filming in 4k@30fps and 1080@240fps on iDevices. I will refer to this as HEVC in this paper.
 - **Most Compatible:** Uses the Advanced Video Codec (AVC) and is also known as MPEG-4 or H.264. This codec can be used for all videos except 4k@30fps and 1080@240fps on iDevices. I will refer to this as AVC in this paper.
- **Record Video:** Determine resolution and frame rates
 - **Lock Camera:** If enabled, it will not allow the user to switch between cameras while recording.

- Record Slo-mo: Determine resolution and frame rates for Slow motion (Slo-mo) videos.
- Record Stereo Sound: If enabled, you will record stereo sound instead of mono.
- Preserve Settings:
 - Camera Mode: “Preserve the last mode, such as Video, rather than automatically reset to Photo” [8].
 - Creative Controls: “Preserve the last used filter, aspect ratio, light, or depth setting, rather than automatically reset” [8].
 - Live Photo: “Preserve the Live Photo setting, rather than automatically reset to Live Photo turned on” [8].
- Use Volume Up for Burst: You can use the top volume button to capture photos in bursts.
- Scan QR codes: Allows you to scan QR codes while using the camera.
- Grid: Enables a 3x3 grid over the user’s screen to help with capture images using rule of thirds.
- Mirror Front Camera: This flips the image captured on your front camera to appear as you see yourself in the mirror instead of flipped like it usually is.
- Prioritize Faster Shooting: Intelligently adapt image quality when rapidly pressing the shutter.
- Smart HDR: Allows you to capture high dynamic range images and the device determines how to compile it [9].
- Keep Normal Photo: “Smart HDR intelligently blends the best parts of separate exposures into a single photo. Save the normally exposed photo in addition to the HDR version” [8].

- While it is not found in the Camera settings section, GPS settings for the camera are found in *Settings -> Privacy -> Location Services -> Camera*. You have the option to enable precise location. According to the note under the setting, “[enabling the precise location setting] allows apps to use your specific location. With this setting off, apps can only determine your approximate location” [8].

Some of the options above apply only to the photo taking capabilities and do not apply to video recording. In my testing, I found that the video format, video resolution and frame rate, slo-mo resolution and frame rate, stereo sound, and GPS location stood out as options that affected video recording. I was unaware if the HDR option would change anything in the video recording metadata, so I decided to test it along with the GPS location. I recorded four different videos using different parameters and options using iOS 14.0:

- 1st video: No GPS, No HDR, 4k, 60fps
- 2nd video: GPS, No HDR, 4k 60fps
- 3rd video: No GPS, HDR, 4k 60fps
- 4th video: No GPS, No HDR, No Stereo Sound, 4k, 60fps

After reviewing the metadata and comparing them to each other, I found that the GPS location does affect the metadata while the HDR setting does not.

Photos Application Editing Capability

iOS 12, 13, and 14 native Photos application allow users to edit videos taken on their device. This can be done in 2 ways. You can remove parts of the beginning or end of a recorded video. This is referred to as “trimming”. Trimming in the native Photo application was introduced in iOS 12.0. Once you have trimmed your file, you have the option to save it as a new video or save over the original video.

To trim a video, first go into the native Photo Application. Once you found a video you wish to trim, click the “edit” button in the top left corner of the screen. See the red arrow in figure 2.

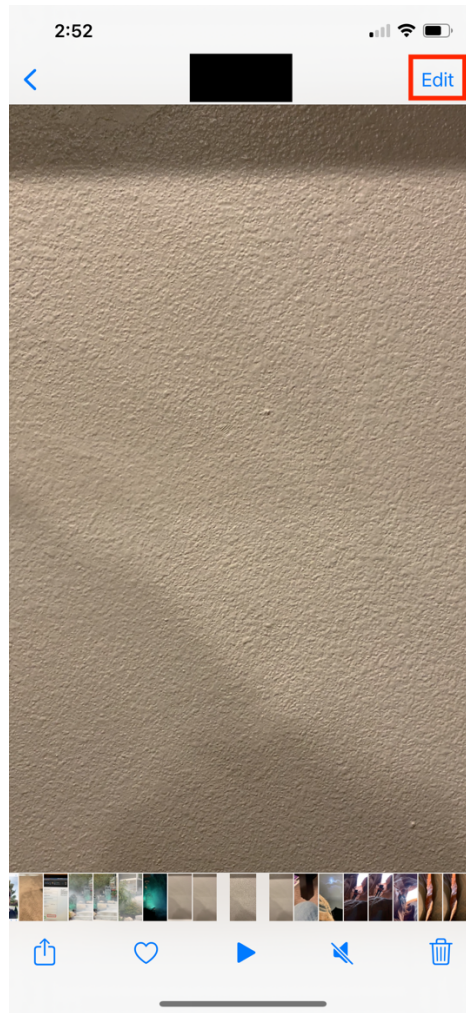


Figure 3. Edit Button in Photo Application

Next, you can slide the yellow bars on the video’s timeline to trim the front and/or end of the video. See the red arrows in figure 4.

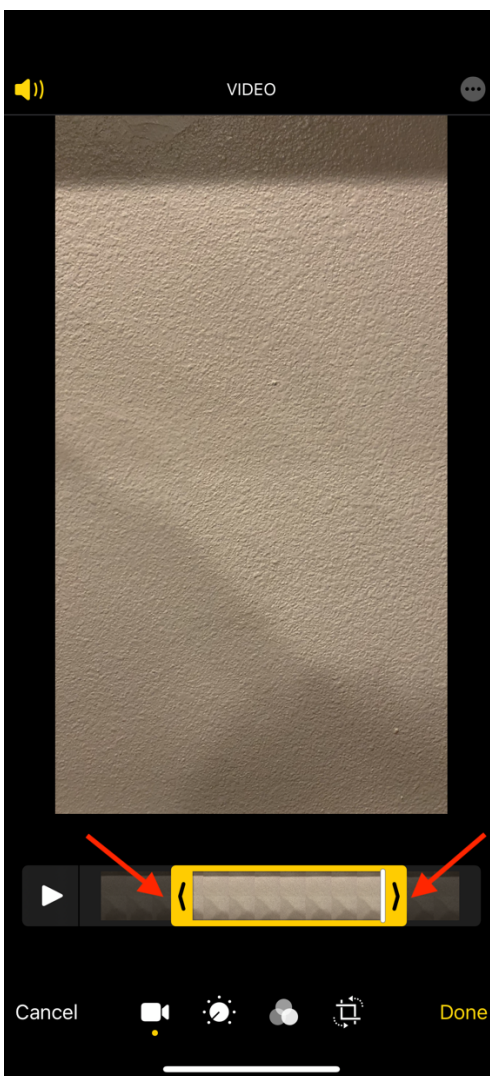


Figure 4. Trimming the Video Using Yellow Arrows on Timeline

You can also remove unwanted areas of an image or video. This is referred to as “cropping”. Cropping in the native Photos application was introduced in iOS 13.0. Once you have cropped your file, you are only able to save this new video over the original video. To do this, go into the Photo application. Once you have found the video you wish to crop, click the edit button in the top right corner. See the red arrows in figure 4.

Next, click the crop button in the bottom middle section. See the red box in figure 5. You still see a crop guide appear on your video. You can touch any of the corners and drag them to

resize the video resolution. See the red arrows in figure 5. Once you have finished making your selection. Click “Done” in the bottom right of the screen.

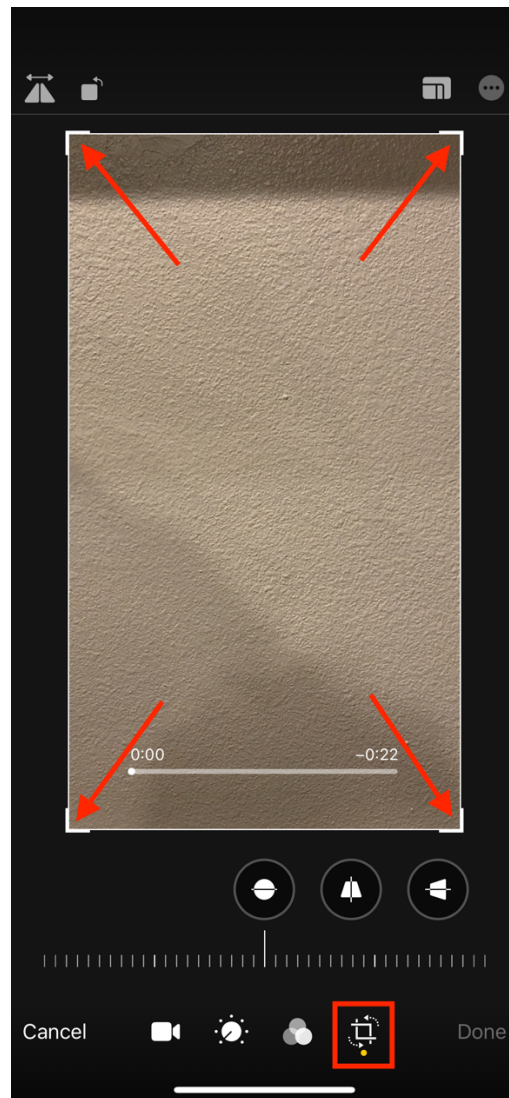


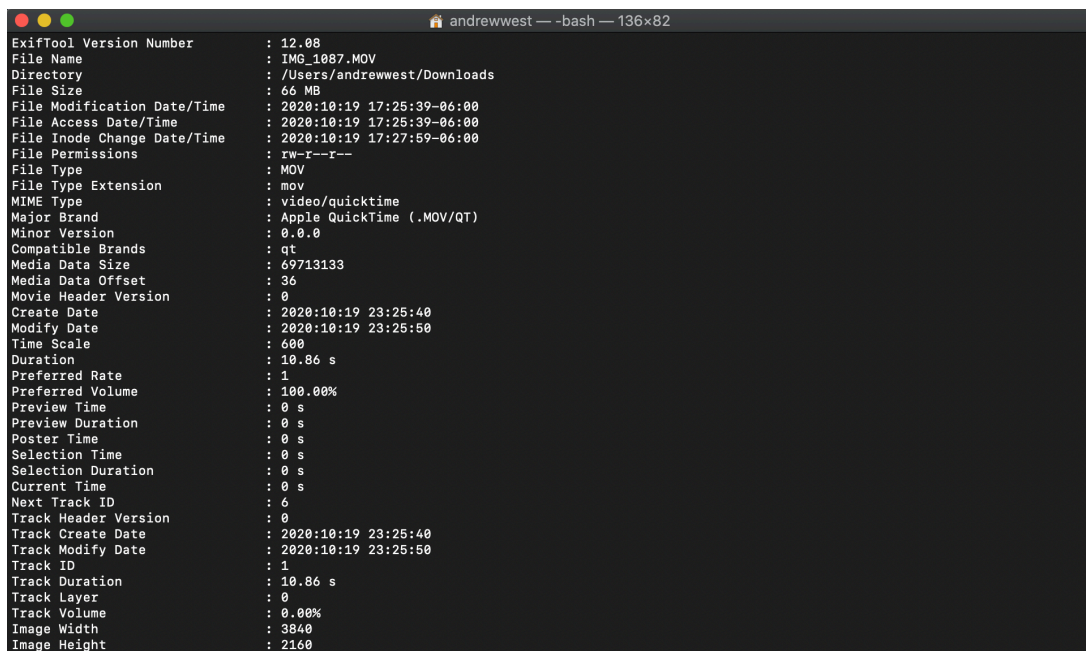
Figure 5. Crop Button and Cropping Guides

After a video has either been trimmed or cropped, iDevice users have two options to save the file. The user has the ability to save the edited video over the original video. This will replace the original file. This feature was introduced in iOS 12 and was the only way to save edited videos. Starting in iOS 13, users also have the ability to save the edited video as a new video.

This will keep the original unedited. This feature allowed the user to option to either save the video as a new file or save over the original.

Comparing Metadata Reader Tools

In order to find the best results from my tests, I needed to find a metadata reader program that was able to extract the most metadata information from a video file. I tested different metadata reader tools including MediaInfo, exiftool, VLC, and FFProbe. Below are screenshots of what metadata are found using the different programs.



```
ExifTool Version Number      : 12.08
File Name                    : IMG_1087.MOV
Directory                    : /Users/andrewwest/Downloads
File Size                    : 66 MB
File Modification Date/Time   : 2020:10:19 17:25:39-06:00
File Access Date/Time        : 2020:10:19 17:25:39-06:00
File Inode Change Date/Time   : 2020:10:19 17:27:59-06:00
File Permissions              : rw-r--r--
File Type                    : MOV
File Type Extension          : mov
MIME Type                    : video/quicktime
Major Brand                   : Apple QuickTime (.MOV/QT)
Minor Version                 : 0.0.0
Compatible Brands             : qt
Media Data Size               : 69713133
Media Data Offset             : 36
Movie Header Version          : 0
Create Date                   : 2020:10:19 23:25:40
Modify Date                   : 2020:10:19 23:25:50
Time Scale                    : 600
Duration                      : 10.86 s
Preferred Rate                : 1
Preferred Volume              : 100.00%
Preview Time                  : 0 s
Preview Duration              : 0 s
Poster Time                   : 0 s
Selection Time                : 0 s
Selection Duration            : 0 s
Current Time                  : 0 s
Next Track ID                 : 6
Track Header Version          : 0
Track Create Date             : 2020:10:19 23:25:40
Track Modify Date             : 2020:10:19 23:25:50
Track ID                      : 1
Track Duration                : 10.86 s
Track Layer                   : 0
Track Volume                  : 0.00%
Image Width                   : 3840
Image Height                  : 2160
```

Figure 6. Screenshot of Exiftool Metadata

General	
Complete name :	IMG_1087.MOV
Format :	MPEG-4
Format profile :	QuickTime
Codec ID :	qt 0000.00 (qt)
File size :	66.5 MiB
Duration :	10 s 863 ms
Overall bit rate mode :	Variable
Overall bit rate :	51.4 Mb/s
Encoded date :	UTC 2020-10-19 23:25:40
Tagged date :	UTC 2020-10-19 23:25:50
Writing library :	Apple QuickTime
com.apple.quicktime.make :	Apple
com.apple.quicktime.model :	iPhone XS Max
com.apple.quicktime.software :	14.2
com.apple.quicktime.creationdate :	2020-10-19T17:25:39-0600

Video	
ID :	1
Format :	HEVC
Format/Info :	High Efficiency Video Coding
Format profile :	Main@L5.1@High
Codec ID :	hvc1
Codec ID/Info :	High Efficiency Video Coding
Duration :	10 s 863 ms
Bit rate :	51.2 Mb/s
Width :	3 840 pixels
Height :	2 160 pixels
Display aspect ratio :	16:9
Rotation :	90°
Frame rate mode :	Variable
Frame rate :	60.000 FPS
Minimum frame rate :	60.000 FPS
Maximum frame rate :	66.667 FPS
Color space :	YUV
Chroma subsampling :	4:2:0
Bit depth :	8 bits
Bits/(Pixel*Frame) :	0.103
Stream size :	66.3 MiB (100%)
Title :	Core Media Video
Encoded date :	UTC 2020-10-19 23:25:40
Tagged date :	UTC 2020-10-19 23:25:50
Color range :	Limited
Color primaries :	BT.709
Transfer characteristics :	BT.709
Matrix coefficients :	BT.709
Codec configuration box :	hvcC

Figure 7. Screenshot of Mediainfo Metadata

After comparing one video over these metadata readers I found that Mediainfo and exiftool produced a lot of metadata and VLC and FFProbe provided a small amount. There were minimal differences between MediaInfo and exiftool but produced the same metadata results. For the exact results, see the appendix. I decided to use with Mediainfo as my metadata tool for reviewing video files. Other metadata tools may parse each file differently and may use different language when reporting the results. However, I will try to refer to each metadata as the Apple Developer standards calls it. If there is no name for it in Apple Developer standards, I will refer it to as Mediainfo reads it as.

CHAPTER III

METHODOLOGY

In proper forensic practices, it is required that the evidence should never be changed in anyway. By doing so, it would degrade the integrity of the evidence and possible change it. To avoid this, the Scientific Working Group of Digital Evidence (SWGDE) has defined best practices for handling digital evidence. In this methodology, I followed the guidelines and procedures as they are set out in SWGDE's Best Practices for Digital Forensic Video Analysis [17]. This includes workflow, examination, and interpretation.

Verifying Transfer Methods

One issue that became present in my initial research is that at the during the time of researching this issue, Apple had recently released iOS 13.5.1. Apple, in an effort to keep their iDevices secure, stops signing older iOS versions due to users jailbreaking their devices by exploiting a flaw in iOS 13.5 [10]. This meant that I would only be able to take videos using iOS 13.5.1 and forward. Finding iDevices still on older versions would be next to impossible. I researched multiple methods on how to install older versions on my test iPhone. The only way I found in my research was to jailbreak my test iPhone and install an iPod Software (IPSW), a file used to install iOS firmware, for each version from various websites [11].

This method has two issues. First, jailbreaking a phone could potentially change how the native Photos application interacted with the video file when edited. Second, I would be unable to authenticate the legitimacy of the IPSW I would use. This would defeat the possibility of testing video files from a wide variety of iOS versions.

The solution to this issue was to use old videos from previous versions. In order to verify that this process was possible, I would need to verify that the transfer methods used to get those

videos would not alter the file when transferred. In order to ensure that videos used for testing would not be altered when transferred, we need a test to verify its integrity. According to SWGDE, using hash values can help determine if the file has been altered in any way [12]. I decided to use Jacksum, a checksum utility, to verify if the file hashes had changed during transfer. I then performed some tests to see if the hash values were changed over three popular transfer methods including:

1. AirDrop: Apple's wireless transfer method found on iDevices and Macs
2. Image Capture: a native MacOS application used to transfer videos
3. iCloud: Apple's premium cloud storage service

AirDrop and Image Capture Verification

Apple's AirDrop is a method to wirelessly send a copy of images, videos, documents, and more to other iDevices using a peer-to-peer Wi-Fi connection. This feature was introduced with iOS 7 and is available with the then current iDevices including iPhone 5, iPad (4th generation), iPad Pro, iPad Mini, and iPod Touch (5th generation). All iDevices since have had AirDrop and can be found on current versions of iOS, iPadOS, and MacOS [13][14][15]. Another method of transfer is using Image Capture. Image Capture is a native application found in macOS that allows users to transfer videos and images from your iDevices.

To test AirDrop and Image Capture as viable transfer methods, I used the following testing procedure.

1. Using my test iPhone, I recorded a short 10 second video on iOS 14.0.
2. I hooked the test iPhone up to my MacBook Pro using a standard lightning cable.
3. Using the MacBook Pro Image Capture software, I transferred the video I just recorded from the test iPhone to the MacBook Pro.

4. Using AirDrop, I sent the same video to my MacBook Pro for comparison.
5. Using Jacksum, I verified the hashes of the file transferred with Input Capture against the same video file transferred by AirDrop.

Comparing the hashes of the two files, I found that both files had the same hashes. Since there are no differences between the hashes, this shows that transferring videos using AirDrop and Image Capture does not alter the hash or file and is a viable way to transfer videos without compromising the integrity of the video file.

iCloud Verification

Another way to overcome the problem of finding older videos was to search through my past collection of videos. I had a personal collection of videos stored in my personal iCloud from previous years. iCloud is an online backup service provide by Apple that allows you to back up and store your data on your devices including documents, settings, images, videos, files, and more [16]. I had used this service for more than 4 years. I had backed up multiple videos from previous iOS versions.

In order to test using iCloud as a viable transfer and storage method and did not alter stored videos, I followed the following testing procedure:

1. Using the camera application on my test iPhone, I captured a video.
2. Using iCloud's setting to automatically backup videos, I let it back it up to the iCloud account.
3. I then downloaded the file from iCloud to my MacBook Pro.
4. Once the file was received on my test MacBook Pro, I then sent the original video via AirDrop to my MacBook Pro for comparison.
5. Using Jacksum, I hashed the original file and the second file.

Comparing the hashes of the two files, I found that both files had the same hashes. Since there are no differences between the hashes, this shows that transferring and storing videos using iCloud does not alter the hash or file and is a viable way to transfer and store videos without compromising the integrity of the video file. This method also showed us that I would be able to use old videos from previous iOS versions stored on iCloud for testing.

Video Files from Various Past OS

After authenticating that iCloud did not alter stored videos, I found that I had unedited videos from previous OS versions. I transferred the videos to my MacBook Pro to keep as controls before transferring the videos to the test iPhone. Using Mediainfo, I found that I had videos for the following versions and video codecs.

Table 1. iOS Versions and Video Codecs

iOS Version	Codec Used on Video
12.0	HEVC
12.0.1	HEVC
12.1	HEVC
12.1.2	HEVC
12.1.4	HEVC
12.2	HEVC
12.3.1	HEVC
13.0	HEVC
13.1	HEVC
13.1.2	AVC
13.2	AVC
13.2.3	HEVC
13.3	HEVC
13.3.1	HEVC
13.7	AVC, HEVC
14.0	AVC, HEVC

CHAPTER IV

METHOD

List of Equipment and Resources

With the three transfer methods verified all of the videos were gathered together, I was able to begin testing. I used the following equipment and resources to extract, transfer, and

- MacBook Pro: This computer was used for verifying hashes and hex data from the videos. The computer also stores the edited and control videos from the test iPhone.
- Test iPhone: A cleanly wiped iPhone 6s that was and restored to factory settings. Its purpose is to be used to edit the videos transferred from MacBook Pro. It was on iOS 13.5.1.
- iCloud: Personal account used to retrieve stored videos from past iOS versions.
- Jacksum: A checksum software to verify that the file had not been altered.
- 010 Editor: A hexcode editor/viewer to verify what information could be found in the video file.
- Mediainfo: A metadata viewer that allowed me to see what metadata information could be found in the video file.

Process for Trimming the Video

For each video, I did the following:

1. Using the edit function in Photos application, I trimmed the beginning of the video down to 10 seconds.
2. I then saved the video as a new video.
3. I went back to the original video; I trimmed the beginning of the video down to 10 seconds again.

4. I then saved the video over the video.
5. I transferred the two videos to my MacBook Pro via AirDrop for comparison.

I repeated this process for each video for every iOS version.

Process for Cropping the Video

For each video, I did the following:

1. Using the edit function in Photos application, I cropped the video down to around half the vertical size.
2. I then saved the video over the video.
3. I transferred the video to my MacBook Pro via AirDrop for comparison.

I repeated this process for each video for every iOS version.

CHAPTER V

RESULTS

Reading Metadata

The Scientific Working Group on Digital Evidence has a best practice for Digital Forensic Video Analysis. Best practice states that Metadata examination is a valid form of authenticating a video file [17]. Using the program, Mediainfo, I found what metadata could be found in the files. I then compiled the data into a single excel file to compare all the file's metadata together.

Changes Found in the Metadata

Small Factors and Observations

In my testing, I found a couple of small factors that appeared in a couple of files:

- When editing videos in iOS 12.0's Photo Application, after trimming a video file on Photos, the file may have the word, "TRIM" attached to the filename when it is saved as a new file.
- iPhones have the ability to embed your GPS locations into the video file. GPS locations on videos are an optional feature. This can also be helpful when authenticating the video.
 - You can check if this setting is enabled by going to *Settings -> Privacy -> Location Services -> Camera*.
 - If the option, "Never" is selected, photos will not contain any GPS data. If the option, "While Using the App" is selected, photos will contain GPS data.

Necessary Metadata

There is some necessary metadata that is found in every video recorded on iOS 12, 13, and 14. This includes the iOS version the video was recorded on, the make and model of the

device, the files encoded date, the file tagged date, the video codec used, and the format profile according to bit rate of file.

Clean Aperture Width and Height

When a video is cropped, two new metadata fields are created called, “Clean aperture width” and “Clean aperture height”. These two new metadata names are what Mediainfo reports them as, however they are the new width and heights of the cropped videos. An original video will not have these metadata fields in them. You can find this new metadata field in the hex code by searching for, “clap”. Figure 8 demonstrates a file that has been cropped and has, “clap” in it’s hex data. Figure 9 demonstrates that same uncropped file that has not been cropped. It does not include, “clap” in its hex data.

```

É....colnclc...
.....(clap...š.
.....8.....
.....
..@sgpd....tscl.
.....`.....°..
...{.....`..
.°.....{.....
..!sgpd....sync.
.....sgp
d....tsas.....
.....Äsbgp...
.sync...V.....

```

Figure 8. Screenshot of “Clap” Clean Aperture Data in Cropped File’s Hex Code

```

á..0a.%.colrn
clc.....
hsgpd....tscl...
.....³€...
.{.....³
€.....{.....`
.°.....{.....
...`.....°.....{..
.....sgpd...
.sync.....
...sgpd....tsas.
.....Äs
bgp....sync.....

```

Figure 9. Screenshot of Missing Clean Aperture Data in Uncropped File’s Hex Code

Below is a sampling from the testing results with the cropped (C) versions on the left and the original (O) versions on the right.

Filename	✗ 12.0 OG.MOV	✗ 12.0 C.mov	✗ 12.1 OG.MOV	✗ 12.1 C.mov
General				
Clean aperture width		1 162 pixels		864 pixels
Clean aperture height		1 080 pixels		1 080 pixels

Filename	✗ 12.1.2 OG.MOV	✗ 12.1.2 C.mov	✗ 12.1.4 OG.MOV	✗ 12.1.4 C.mov
General				
Clean aperture width		1 100 pixels		1 040 pixels
Clean aperture height		2 160 pixels		2 160 pixels

Figure 10. Clean Aperture Width and Height Comparison

Frame Resolution and Frame Rate

Each iDevice may use a different filming resolution and frame rate. The chart located in the appendix shows what devices are able to run iOS 12, 13, and/or 14 along with what filming resolutions and frame rates are possible. If the video is different than what iPhone is the metadata, then it is not original to that phone. You can find this chart for reference in Appendix A.

HEVC files use a variable frame rate while recording, but the iDevice tries to keep the frame rate as close to the chosen frame rate. The frame rate may dip up and down by a little bit (+/- 2 frames). For example, filming in 60fps may result in a 59.940 frame rate. If the frame rates dip way below what it should be, that is indication that the file may be edited. Due to time constraints, I was unable to fully test why frame rates dip. It may be a matter of CPU usage or something else. This would be an area of study in the future. AVC files use a constant frame rate and should never change. If it is outside of what that phone is capable of, then it has been edited or is not from that original phone.

Encoded and Tagged Date

MediaInfo uses the terms, encoded date and tagged date to label the start and the end of the file. The encoded date is the beginning time of the video file or when the user pressed the

record button. We will refer to this as the start time going forward. The tagged date is the end time of the video file or when the user stopped the recording. This will be referenced as the end time.

Filename	✖ 12.0 OG.MOV	✖ 12.0 NF.MOV	✖ 12.0 SO.mov
General			
Encoded date	UTC 2018-10-03 14:48:46	UTC 2020-09-11 21:21:42	UTC 2020-09-11 21:22:47
Tagged date	UTC 2018-10-03 14:49:07	UTC 2020-09-11 21:21:42	UTC 2020-09-11 21:22:47

Figure 11. Start and End Date Comparison for Trimmed Video Files

Figure 11 is a small snippet of the test results and shows the start and end dates for an original and trimmed file that have been saved as a new file and saved over the original file. Every instance of the start and end time and dates of the file will be different in an unedited file with no indication of a previous date. Cropped video files behave differently. Unlike trimmed videos, cropped video's start and end times appear as normal encoded and tagged dates. However, the new start and end date and times are when the edited video was saved. This is shown in Figure 12.

Filename	✖ 12.0 OG.MOV	✖ 12.0 C.mov	✖ 12.1 OG.MOV	✖ 12.1 C.mov
General				
Encoded date	UTC 2018-10-03 14:48:46	UTC 2020-10-01 00:30:48	UTC 2018-11-20 20:07:42	UTC 2020-10-01 00:31:08
Tagged date	UTC 2018-10-03 14:49:07	UTC 2020-10-01 00:31:08	UTC 2018-11-20 20:08:07	UTC 2020-10-01 00:31:23

Figure 12. Start Date and End Date Comparison for Cropped Video Files

In cropped video files, the start date will be the time the file has edited and saved. The end date will be the duration of the video added onto the new start time of the file. This applies to both HEVC and AVC files.

Source Duration

The duration, or source duration in MediaInfo, is the duration of the video content in the file. The duration of the file is not created from a subtraction of the encoded and tagged date. It is

a four-byte independent factor found in the “mdhd” media header atom [20]. However, the duration should follow this equation:

$$Start\ Date - End\ Date = Source\ Duration$$

If the file’s metadata does not follow this equation, then the file’s start date, end date, or duration has been edited.

Filename	✗ 12.0 OG.MOV	✗ 12.0 NF.MOV	✗ 12.0 SO.mov
General			
Duration	21 s 347 ms	2 s 87 ms	2 s 868 ms
Encoded date	UTC 2018-10-03 14:48:46	UTC 2020-09-11 21:21:42	UTC 2020-09-11 21:22:47
Tagged date	UTC 2018-10-03 14:49:07	UTC 2020-09-11 21:21:42	UTC 2020-09-11 21:22:47

Figure 13. Duration of Video Files

In Figure 13, it shows how the original files for iOS 12 and 12.1 follow this equation. We see that because the edited files show the same start and end date, they do not follow this equation. This applies to both HEVC and AVC files but only for trimmed videos.

GOP and Reference Frames

Since AVC files is the only codec on an iDevice that uses constant frame rates, it is the only codec that will produce GOP and Reference Frames in its metadata. Group of Pictures (GOP) or GOP Structure specifies what order the interframes are connected. The GOP Structure is referred to by two numbers, for example M=1 N=15 [18][19]. The first number refers to the distance between the two anchor frames (I or P) [18][19]. The second number refers to the distance between the full images (I frame) [18][19]. An original video will contain GOP formats in its metadata while an edited video that has been trimmed will not as demonstrated in Figure 14.

Filename	✖ 13.7 OG.MOV	✖ 13.7 NF.mov	✖ 13.7 SO.MOV
General			
Format settings, Reference frames	1 frame	1 frame	1 frame
Format settings, GOP	M=1, N=15		

Figure 14. GOP and Reference Frames

Source Stream Duration

HEVC is the only codec on an iDevice that uses a variable frame rate. When a HEVC file is edited, it goes through the compression algorithm again and creates a new metadata section called, “source stream size” in Mediainfo. This metadata field indicates that the file has been trimmed and saved as either a new file (NF) or over the original file (SO). If the file is an original, as labeled as “OG”, it will not contain this metadata field as shown in Figure 15.

Filename	✖ 12.0 OG.MOV	✖ 12.0 NF.MOV	✖ 12.0 SO.mov
General			
Source duration		2 s 120 ms	2 s 885 ms
Source stream size		4.69 MiB (99%)	6.32 MiB (99%)

Filename	✖ 12.1 OG.MOV	✖ 12.1 NF.mov	✖ 12.1 SO.MOV
General			
Source duration		1 s 485 ms	1 s 17 ms
Source stream size		2.48 MiB (98%)	1.95 MiB (98%)

Figure 15. Source Stream Duration

Hoov Movie Data Header

According to Apple’s Quicktime Developer guide, Quicktime video file hex data is organized into tables or atoms [19]. These atoms can tell us different information such as tagged date, encoded date, duration, title, etc. This is where the file’s metadata information is stored. Most of the atoms are stored at the end of the file under the atom named, “moov”. However, in the media data, which stores the video frames, there is a hidden table titled, “hoov”. The “hoov” atom contains duplicate metadata information that are also found in other media

headers. Below are some of the possible metadata information that can be found in the “hoov” media header.

Table 2. “hoov” Media Header Information

hoov	Prof	hdlr	dhlralisappl	stsd
mvhd	Enof	Hdlrvideoappl	Core Media Handler	hvc1
Trak	edts	Core Media Video	dinf	HEVC
Tkhd	Elst	minf	dref	hvcC
Tapt	mdia	vmhd	alis	
Clef	mdhd	hdlr	stbl	

This ”hoov” media header atom is only found in unedited files. It appears towards the middle of the file data but the placement is not consistent. Anytime a file is edited in the Photo application and saved, the hoov atom is deleted. Using a hex viewer, you can search the file using the following term, “hoov”. If the “hoov” is not found in the file, it has been edited. This applies to both HEVC and AVC files.

The image shows a hex dump of a video file header. The hex data is displayed in columns, with the ASCII representation on the right. The 'hoov' atom is clearly visible in the ASCII column, starting with 'hoov' and followed by various metadata fields like 'mvhd', 'edts', 'tkhd', 'tapt', and 'clef'.

Figure 16. “hoov” Media Header Atom in Unedited Video File

Decision Tree

In an effort to make a streamlined process for determining if a video is consistent with the iOS versions tested, I have created a decision tree using the information previously discussed.

See figure 17 for the decision tree.

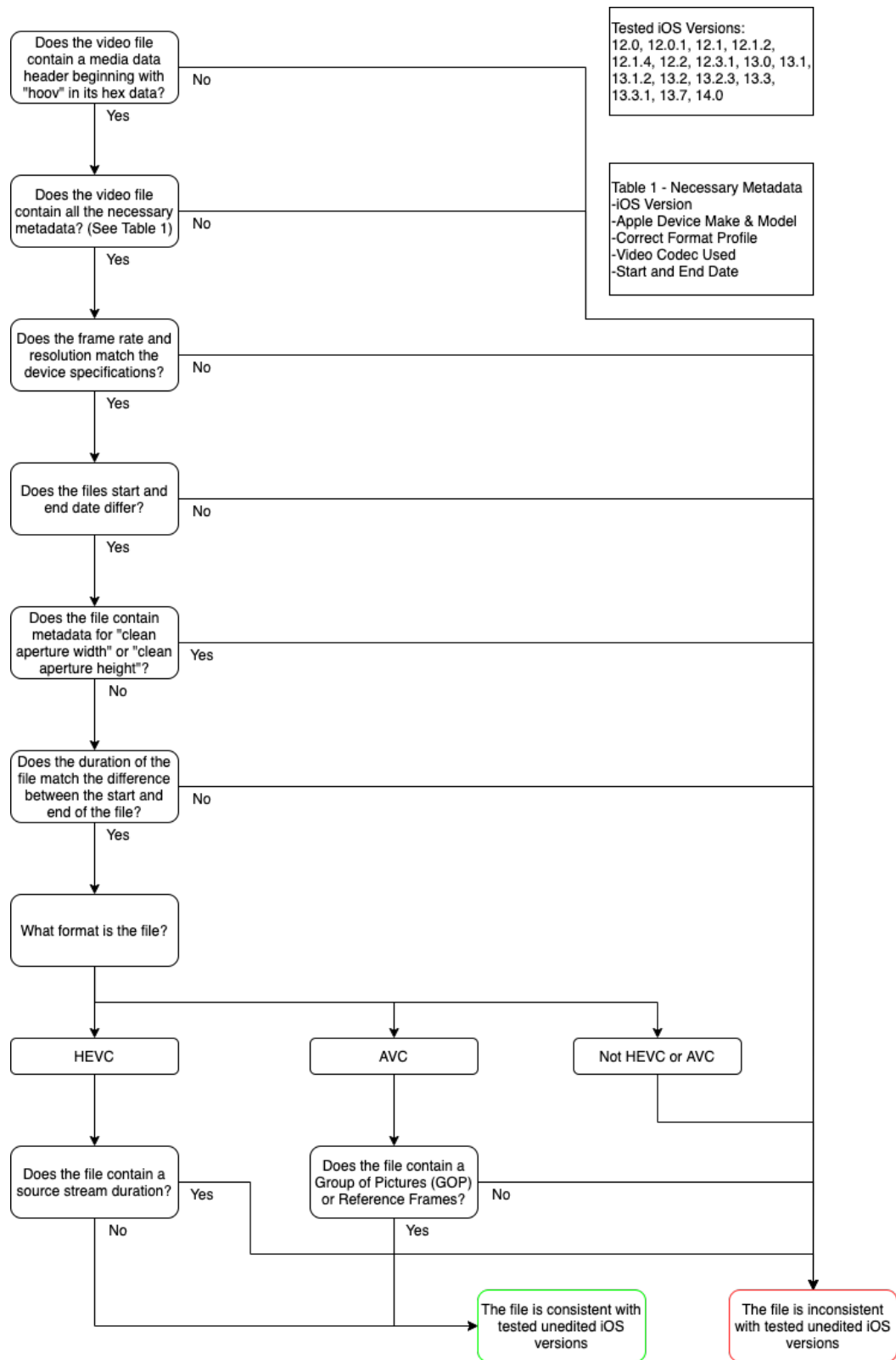


Figure 17. Decision Tree for Verifying Consistency with Tested Unedited iOS Versions

CHAPTER VI

CONCLUSIONS

When comparing unedited videos shot on iOS 12, 13, and/or 14 to the trimmed and cropped versions of the videos, there are numerous differences in the metadata and hex data. These differences can be adding, deleting, or changing a metadata value from its original state. The metadata values that are altered include:

- Clean Aperture Width and Height
- Frame Rate
- Encoded Data (Start Date)
- Tagged Date (End Date)
- Source Duration for HEVC files
- GOP and Reference Frames for AVC files
- “Hoov” Movie Data Header

These changes can help a forensic examiner be able to understand if a video file from iOS 12, 13, or 14 is consistent with unedited video that has not been edited on the native Photo application.

This method of using metadata and hex code to verify a video file is not a catch all solution. The values discussed in this paper can be easily manipulated by changing a file’s metadata and hex code. In this case, it is highly recommended that other forms of authentication be used to validate the video file in question. This form of authentication can support forensic examiners conclusions and findings.

With the iPhone and iPad still being a success in the smartphone and tablet market, it is evident that a high majority of consumers will continue to use these devices. Because of the ease

of editing a video on the native Photo application, it is necessary to use this detection methods to determine if video files recorded on iDevices have been edited using the application.

Possible Future Topics of Research

To continue in being proactive in researching how we can detect if digital media files have been edited on these devices, possible future topics of research include the following and are not limited to monitoring for changes in future iOS updates, investigating if the audio in a video file changes when a video file is trimmed or cropped, other iOS 12, 13, and 14 updates that were not tested in this study, authentication of video files edited on the native Photos application found on the Mac, and what other metadata information can be found within the “hoov” media header. I also mentioned within this study that due to time restraints, I was unable to test why frame rates may dip when using the HEVC codec. This and previous topics are studies that will help to further develop authentication for video files recorded using Apple’s iOS.

REFERENCES

1. Tankovska, H. 2020. *Apple iPhone Sales by Year 2007-2018*. September 21. www.statista.com/statistics/276306/global-apple-iphone-sales-since-fiscal-year-2007/.
2. Nations, Daniel. 2020. *Did You Know This Many iPads Had Been Sold?* September 11. www.lifewire.com/how-many-ipads-sold-1994296.
3. Sui, Linda. 2020. *Apple Becomes World's No.1 Smartphone Vendor in Q4 2019*. January 30.
4. 2019. *Shot on iPhone 11 Pro - Snowbrawl*. Directed by David Leithch. Produced by Apple Inc. YouTube.
5. "iOS 12 is Available Today." September 17th, 2018. Apple Newsroom. <https://www.apple.com/newsroom/2018/09/ios-12-is-available-today/>
6. Warren, Tom. 2019. "IOS 13 Is Now Available to Download." The Verge. September 19, 2019. <https://www.theverge.com/2019/9/19/20874064/apple-ios-13-download-available-now-ipad-iphone>.
7. "IOS 14 Is Available Today." September 16, 2020. Apple Newsroom. <https://www.apple.com/newsroom/2020/09/ios-14-is-available-today/>.
8. Apple Inc. (2020) Settings (version 14.0) [Mobile app]
9. "Adjust HDR camera settings on iPhone," *Apple Support*. <https://support.apple.com/guide/iphone/adjust-hdr-camera-settings-iph2cafe2ebc/ios>.
10. "Apple Has Just Patched the Recent IOS 13.5 Jailbreak | TechCrunch." June 1, 2020. <https://techcrunch.com/2020/06/01/apple-patch-jailbreak/>.
11. "IPSW File - How Do I Open a .Ipsw File? [Step-by-Step]." n.d. FileSuffix.Com. <https://www.filesuffix.com/en/extension/ipsw>.
12. Scientific Working Group on Digital Evidence. 2019. "SWGDE Practical Considerations for Submission and Presentation of Multimedia Evidence in Court." Google Docs. September 29, 2019. https://drive.google.com/file/d/1hrhGcHxYDvg5L8GExnLheFXrB1DoD9y/view?usp=embed_facebook.
13. "How to Use AirDrop on Your iPhone, iPad, or iPod Touch." n.d. Apple Support. <https://support.apple.com/en-us/HT204144>.
14. "Use AirDrop on Your Mac." n.d. Apple Support. <https://support.apple.com/en-us/HT203106>.

15. "What Is AirDrop and How Does It Work?" Lifewire. <https://www.lifewire.com/what-is-airdrop-how-does-it-work-1994512>.
16. "What Is iCloud?" n.d. Apple Support.. <https://support.apple.com/guide/icloud/what-is-icloud-mm74e822f6de/icloud>.
17. SWGDE. 2018. "SWGDE Best Practices for Digital Forensic Video Analysis." *Scientific Working Group on Digital Evidence*. <https://drive.google.com/file/d/1vXMav2QS8N5iiC7UcW6ZoTsH0OHI7FQi/view>.
18. Apple Inc. 2012. "Compressor 4 User Manual." help.apple.com/compressor/mac/4.0/en/compressor/usermanual/.
19. SWGDE. 2018. "SWGDE Technical Overview of Digital Video Files." *Scientific Working Group on Digital Evidence*. https://drive.google.com/file/d/1pi_6wx67TJ1rfyDknDe0z5m-NK85hW9O/view.
20. Apple Inc. "QuickTime File Format Specification." September 13. developer.apple.com/library/archive/documentation/QuickTime/QTFF/QTFFChap2/qtff2.html.
21. Supported iPad Models - iPadOS 13." Apple Support. <https://support.apple.com/guide/ipad/supported-models-ipad213a25b2/13.0/ipados/13.0>.
22. Supported iPad Models - iPadOS 14." Apple Support. <https://support.apple.com/guide/ipad/supported-models-ipad213a25b2/14.0/ipados/14.0>.
23. "Supported iPhone Models - iOS 12." Apple Support. <https://support.apple.com/guide/iphone/supported-models-iphe3fa5df43/12.0/ios/12.0>.
24. Supported iPhone Models - iOS 13." Apple Support. <https://support.apple.com/guide/iphone/supported-iphone-models-iphe3fa5df43/13.0/ios/13.0>.
25. Supported iPhone Models - iOS 14." Apple Support. <https://support.apple.com/guide/iphone/supported-iphone-models-iphe3fa5df43/14.0/ios/14.0>.

APPENDIX

Apple Device iOS Compatibility, Camera Resolution, and Frame Rate

[21][22][23][24][25]

	OS Compatibility			Camera Resolution and Frame Rate									
				4k			1080p				720		
Apple Device	iOS 12	iOS 13	iOS 14	24	30	60	30	60	120	240	30	120	240
iPhone 5S	✓	X	X				✓					✓	
iPhone 6/6 Plus	✓	X	X					✓					✓
iPhone 6S/6S Plus	✓	✓	✓		✓			✓	✓				✓
iPhone SE (1st generation)	✓	✓	✓		✓		✓	✓	✓				✓
iPhone SE (2nd generation)	Phone Not Yet Available	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPhone 7/7 Plus	✓	✓	✓		✓		✓	✓	✓				✓
iPhone 8/8 Plus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPhone X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPhone XS/XS Max	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			

iPhone XR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPhone 11	Phone Not Yet Available	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPhone 11 Pro	Phone Not Yet Available	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPhone 11 Pro Max	Phone Not Yet Available	✓	✓	✓	✓	✓	✓	✓	✓	✓			
iPod Touch (6th generation)	✓	X	X				✓				✓	✓	
iPod Touch (7th generation)	✓	✓	✓				✓				✓	✓	
iPad Air (1st generation)	✓	Moved to iPadOS 13	iPadOS 14				✓				✓		
iPad Air 2	✓	Moved to iPadOS 13	iPadOS 14				✓				✓	✓	
iPad Air (3rd generation)	✓	Moved to iPadOS 13	iPadOS 14				✓				✓		
iPad (5th generation)	✓	Moved to iPadOS 13	iPadOS 14				✓				✓	✓	

iPad (6th generation)	✓	Moved to iPadOS 13	iPadOS 14				✓				✓	✓	
iPad Mini 2	✓	Moved to iPadOS 13	iPadOS 14				✓				✓		
iPad Mini 3	✓	Moved to iPadOS 13	iPadOS 14				✓				✓		
iPad Mini 4	✓	Moved to iPadOS 13	iPadOS 14				✓				✓		
iPad Mini (5th generation)	✓	Moved to iPadOS 13	iPadOS 14				✓				✓		
iPad Pro (12.9-inch 1st, 2nd, and 3rd generation)	✓	Moved to iPadOS 13	iPadOS 14				✓				✓	✓	
iPad Pro (9.7-inch)	✓	Moved to iPadOS 13	iPadOS 14		✓	✓	✓	✓	✓				✓
iPad Pro (10.5-inch)	✓	Moved to iPadOS 13	iPadOS 14		✓	✓	✓	✓	✓				✓
iPad Pro (11-inch 1st generation)	✓	Moved to iPadOS 13	iPadOS 14		✓	✓	✓	✓	✓				✓