MPEG-4 VIDEO AUTHENTICATION

USING FILE STRUCTURE AND METADATA

by

J. RANDOLPH HALL

B.F.A., Ithaca College, 2002

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

Masters of Science

Recording Arts

2015

This thesis for the Master of Science

degree by

J. Randolph Hall

has been approved for the

Recording Arts Program

by

Catalin Grigoras, Chair

Jeff M. Smith

Jason R. Lewis

Date: November 12, 2015

Hall, J. Randolph (M.S., Recording Arts)

MPEG-4 Video Authentication Using File Structure and Metadata

Thesis directed by Professor Catalin Grigoras

ABSTRACT

The goal of this thesis is to research the file structure of MPEG-4 video files, the contents of the multiple data containers within each file, and the possibilities and limitations of using this information to authenticate a MPEG-4 file. This thesis will impact the forensic science community by showing a method of analysis to examine the meaningful components of a MPEG-4 recording and parse them in order to identify the features of a recording that are consistent with an original recording from the device that created it.

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

Approved: Catalin Grigoras

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

CHAPTER I

INTRODUCTION

The focus of this thesis is to demonstrate a framework of how to authenticate a MP4 video recording based on an analysis of its inherent file structure. MP4 video files are represented by the MPEG-4 Standard and defined in ISO/IEC 14496. The MPEG-4 standard and ISO/IEC 14496 have undergone a number of amendments and additions since its introduction in 1999. The structure of these files is based on the Apple QuickTime container format first published by Apple Computer, Inc. in 2001. The extensible architecture of this file structure has allowed changes to be made within the format over time, while allowing it to remain a viable and useful file format fifteen years after its introduction. In its current form, MP4 files are a popular container of H.264-encoded video, are natively supported in the HTML5 becoming a new standard of web-based video, and represent the majority of video created by consumer cameras and mobile devices.

At its root, the extensible nature of this file format is what allows a given MP4 file to be authenticated as being consistent with the device that was claimed to have created it. In the research for this thesis, a database of sixty-six video recordings was created containing exemplar recordings from a variety of cameras and mobile devices. These recordings were transferred from their respective devices in a forensically sound manner, making sure to preserve the original file structure. By parsing the structure of these files, identifying characteristics can be recognized in their structure as defined by the Apple

QuickTime container format. Due to the inherent design of the file format, there are very few requirements of what containers must be present and how they are configured in any given file. Due to the variety in this structure of containers, identifying characteristics become apparent when comparing the files between manufacturers and models. In addition to the sometimes self-identifying metadata contained within the files, the structure, itself, can be used to authenticate a file as being consistent with the device or to further identify which software was used to handle the file based on how the structure of containers has been modified. Just as physical devices record files in a specific structure of containers, software based manipulation will rearrange the structure of the files they create providing the same basis for identification. The effects of this software interaction vary but no software analyzed for this paper made any attempt to recreate the container structure of the original file.

The National Center for Media Forensics has published proposed frameworks for digital audio authentication[1] and digital image authentication.[2] Conspicuously absent is a framework for the authentication of digital video. There are a number of studies focusing on the authentication of digital video and none of them are more comprehensive than *Forensic analysis of video file formats*, Gloe, et al.[3] This study provides an great deal of detail on specific video file formats, digital cameras, mobile phones, and video editing software, however it stops short of the analysis of MPEG-4 video files based on their file structure. I propose the present study of MPEG-4 file structure format in order to form the basis of a framework for the authentication of digital video.

CHAPTER II

MOTION PICTURE EXPERTS GROUP (MPEG)

The Motion Picture Experts Group (MPEG) was established in 1988 by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). MPEG-1 was their first standard released in 1993 and was defined in ISO/IEC 11172[4]. This first MPEG standard defined a method of encoding moving pictures and audio that would allow playback at the bit rate of a compact disc and at the transmission rate of a T1 line of 1.5 Mbps. MPEG-1 was used primarily in the CD-i video format, Video CD (VCD) format, and in satellite and cable television transmission. The most notable and lasting legacy of the MPEG-1 standard is without question the MPEG-1 Audio Layer III (MP3) audio compression format which remains relevant today.

MPEG-2, defined in ISO/IEC 13818[5], was released in 1996 and made considerable improvements on the MPEG-1 standard. Most notable was the support for a higher transmission bit rate that allowed high definition interlaced video and multi-channel audio streams. MPEG-2 is used in DVD’s, cable television, satellite television, and over-the-air broadcast television. Its hardware is backwards compatible by design so any player capable of playing MPEG-2 encoded data is also capable of playing MPEG-1 data.

MPEG-3, not to be confused with MPEG-1 Layer 3 or MPEG-2 Layer 3, was a standard that never really was. After realizing that the goal of delivering high bit rate streams necessary to provide full 1080p video would be possible with the existing MPEG-2 standard, MPEG-3 was incorporated into MPEG-2 and the standard was shelved.

MPEG-4 OVERVIEW

The MPEG-4 standard has undergone a number of changes since its introduction in 1999. MPEG-4 Part 1, MPEG-4 Part 2, and MPEG-4 Part 3 were the first standards that outlined the file format which was to contain audio and video signals. These standards are defined in ISO/IEC 14496-1[6], ISO/IEC 14496-2[7], and ISO/IEC 14496-3[8]. This structure is based on the Apple QuickTime container format first published in 2001 by Apple, Inc. [9].

A significant amendment to this standard was made in 2003 when MPEG-4 Part 14 was introduced and described in ISO/IEC 14496-14[10]. MPEG-4 Part 14 defined the MP4 file format as it is used today and while there have been many further amendments to the MPEG-4 standard the file structure at its base has remained the same.

MPEG-4 Part 10 defined in ISO/IEC 14496-10[11] introduced H.264/Advanced Video Coding (AVC) in 2003. The storage format for this encoded data was created with MPEG-4 Part 15, defined in ISO/IEC 14496-15[12], released in 2004. H.264 is the video compression standard of the Blu-Ray Disc format. It has also been adopted for online streaming video through services like YouTube, Vimeo, and Apple's iTunes Store. It is used for HDTV over-the-air transmissions, cable, satellite television transmissions, and is the dominant codec used by security system DVR's and digital CCTV systems.

MPEG-4 Part 12 described in ISO/IEC 14492-12[13] defined the ISO base media file format that is at the root of the analysis in this paper. This definition provides the structure for a container file format to store video files locally or transmit them across a network. The structure and contents of these containers is extensible and all registered extensions of the ISO base media file format are maintained by an official registration authority[14]. This provision for the registration of these extensions has existed since MPEG-4 Part 1 was initially released.

CHAPTER III

THE COLLECTION

In creating a database of video files for this thesis, it was important to create a framework by which files could be collected without any opportunities for their structure to be altered when transmitting them from their respective devices. An initial test was performed using a LG G3 mobile phone. In testing the LG G3, a sample video was created and stored on its internal memory. This file was then transferred off of the device using Android File Transfer over a USB connection. The file was then copied to the G3's removable micro SD storage card, sent as an attachment to an email, and synced to another computer using Dropbox. After all of the files had been collected hash values were generated and when compared they all showed matching MD5, SHA-1, and SHA-256 values. In the case of the LG G3 Android device, no transcoding had occurred when transferring a file from the device through any of these techniques.

It should be noted that Dropbox will change the name of the file if using their Camera Upload feature but the structure and contents of the file were not changed. The intra-variability among these methods of retrieving files from their respective devices was zero.

Just because the LG G3 was successful in moving video files off of the device without transcoding them or altering their structure is by no means an endorsement that all other devices will behave in the same fashion. The files not collected personally were created and transmitted using a clear set of guidelines established in order to preserve the originality of the files. When it was not possible to perform such an exhaustive test or when access to the device was not possible, the properties of the files were examined to determine if they had been transcoded in some way to alter their format from the published specifications of their respective device. Consumer cameras and their removable media posed no unexpected challenges in collection. The Android devices, represented in this database, all transmitted files without any modifications using any of the techniques mentioned. While the collection and study of Apple QuickTime files is outside the scope of this paper, it should be noted that the Apple devices examined for the sake of comparison would by default transcode their video files to a much lower quality when attached to an email message. The original files could be retrieved from the device using Dropbox but no further testing was performed on these devices.

In collecting these files, it was worth considering how the average user would share their videos or how these files would most likely and most easily moved off a mobile device with no availability of external storage. Once configured, the ease of Dropbox synchronization is undeniably simple however the two most obvious and ubiquitous choices were moving files via email and MMS messages. As previously observed, an emailed video would retain its original structure on the Android devices examined. In the case of transmitting via MMS message, the Android device transcodes the original file due to size limitations. Once the methods of collecting the video data were validated the most common means of collecting the videos from their respective devices was via email attachments.

When collecting video samples for the database of files to be examined, it was important to create multiple samples from each device. Modern mobile devices have the capabilities to record video at a wide range of resolutions and frame rates; it was important to collect the data from these devices using each of their possible recording modes. It was also important to collect multiple samples of each possible mode so that any variability within a single given device could be identified and investigated further. This behavior was not observed in any of the devices examined.

CHAPTER IV

ANALYSIS

In order to manually parse a MP4 file, it is important to understand the container-based nature of the file itself. The structure of these files is based entirely on the Apple QuickTime File Format Specification[15]. Apple refers to this fact openly in the documentation of their QuickTime standard and states clearly that the primary difference between QuickTime and MPEG-4, “An atom, as described in this document, is functionally identical to a box, as described in the ISO specifications for MPEG-4 and Motion JPEG-2000. An atom that includes version and flags fields is functionally identical to a full box as defined in those specifications.” Conversely, the ISO/IEC 14496-12:2005(E) publication points out that in the first publication of their specification a ‘box’ was referred to as an ‘atom’. For the purposes of this paper, we will refer to these containers as ‘boxes’ as in ISO/IEC 14496-12:2005(E). These boxes act as individual containers or as containers of additional containers nested inside one another.

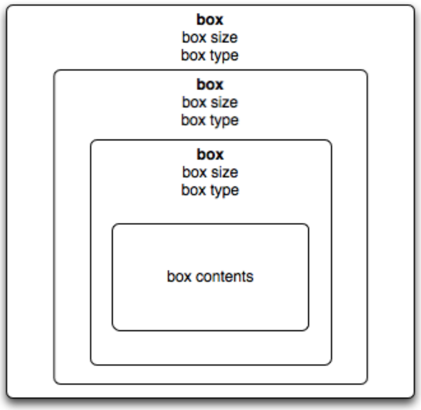


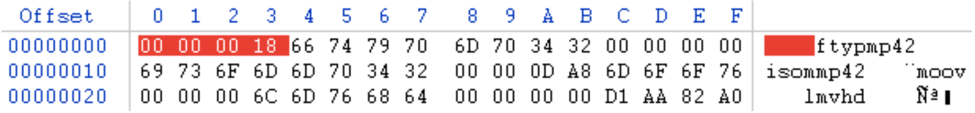
Figure 1. MPEG-4 Box Structure

Each of these boxes begins with an unsigned 32-bit or 64-bit integer in big endian format that defines the size of the box itself. The vast majority of boxes use the 32-bit integer but there are examples of 64-bit sizes in the data surveyed for this paper: a box that is simply so large that it requires a 64-bit integer to represent its size[13], and a series of Universally Unique Identifiers. If the size of the box is 0x00 then the contents of the box extend to the end of the file.[13]

For the purposes of parsing the MPEG-4 boxes all byte size values will be described in hexadecimal values using the prefix ‘0x’ where 0x00=0 bytes, 0x10=16 bytes, 0x20=32 bytes, etc.

The File Type Box

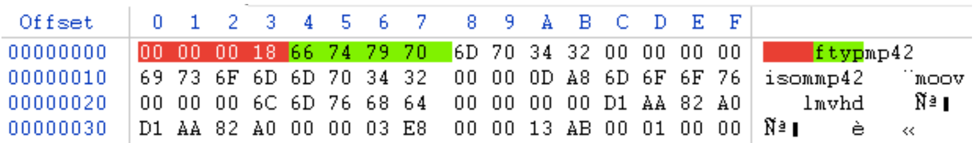
In this example file, the first four bytes represent the size of the box: 0x18 bytes. This measurement includes the bytes used to represent the size of the box itself.



```
 Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42    ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▮
```

Figure 2. MPEG-4 Box Size

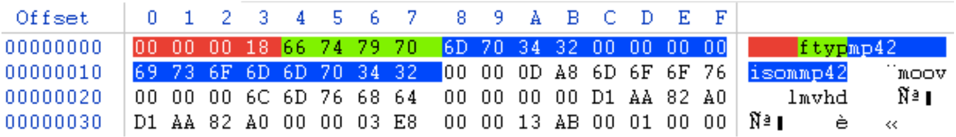
The next four bytes define the type of box. In this example, the first box of the file is ‘ftyp’, a File Type Box. The ISO specification requires this box to exist as early as possible in the file. In the files examined for this paper, it was always the first box in each sample. There can be only one ‘ftyp’ box per file and it must exist in order for the file to meet the ISO specification. The ‘ftyp’ box must also exist at the top level of the file. The File Type Box allows a given file to define compatibility with multiple standards if applicable. In this case, the box contents contains ‘mp42’, ‘isom’, and a second ‘mp42’.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▌
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▌    è    «
```

Figure 3. MPEG-4 Box Type

In this example, the first ‘mp42’ used as a major brand identifier, referring to the use of the Microsoft MPEG-4 codec. The 0x00 at offsets 0x0C through 0x0F act as a placeholder for any identifiers that would be used to define the minor version of the major brand of this file. ‘isom’ and the second ‘mp42’ identify what are referred to as the compatible brands of this File Type Box. In this example, the standards identified in the ‘ftyp’ box are complimentary. In the event where the audio or video were to not follow the ISO standard, the file types would be defined so that a decoder would correctly handle the data for decoding and playback.

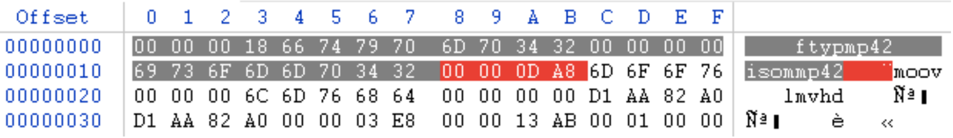


```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▌
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▌    è    «
```

Figure 4. MPEG-4 Box Contents

The Movie Box

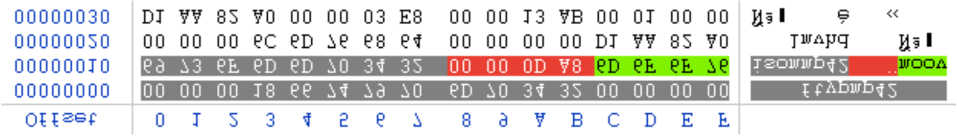
The next four bytes of our file contain the box size for our next box: 0x0DA8.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▌
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▌    è    «
```

Figure 5. Movie Box Size

The four bytes following that define the box: moov.



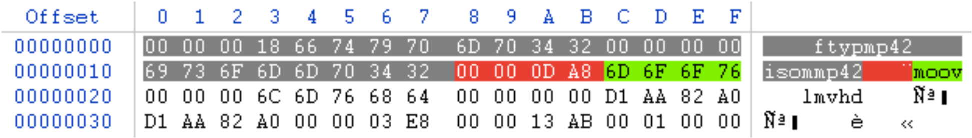
```
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▮     è   «
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▮
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42    moov
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
```

Figure 6. Movie Box Type

‘moov’ identifies this box as a Movie Box. The Movie Box contains the metadata of the file represented in additional boxes. In this example, the moov box contains 3496 bytes, it is significantly larger than the ‘ftyp’ box and contains all of the identifying information describing the contents of the video file. The structure and contents of these metadata boxes are at the root of building a framework to authenticate the file. ‘moov’ is a top-level box that must exist and there can be only one box in order for the file to meet the ISO specification. There are forty-two nested boxes inside this ‘moov’ box but the one of most forensic interest is ‘mvhd’, the Movie Header box.

The Movie Header Box

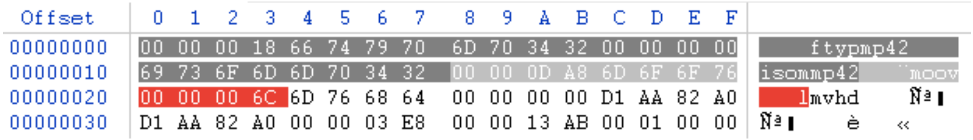
In the research for this paper, the variability in the positioning of the MPEG-4 boxes provided a method to identify a file based on the order and organization of the data containers themselves.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42    moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▮
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▮     è   «
```

Figure 7. MPEG-4 Nested Box Size

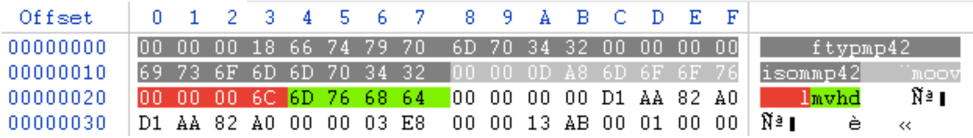
To begin parsing the ‘moov’ box which is 0x0DA8 bytes, there are no immediate contents in this box; instead there is a four byte string identifying the size of another box.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▌
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▌   è   «
```

Figure 8. Movie Header Box Size

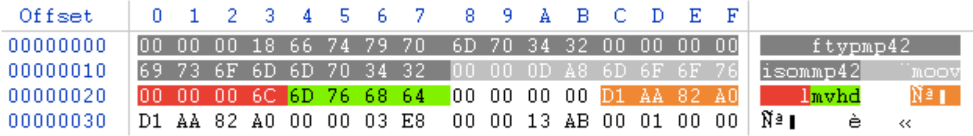
Measuring 0x6C bytes in length this is the first example of a nested box: ‘mvhd’.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▌
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▌   è   «
```

Figure 9. Movie Header Box Type

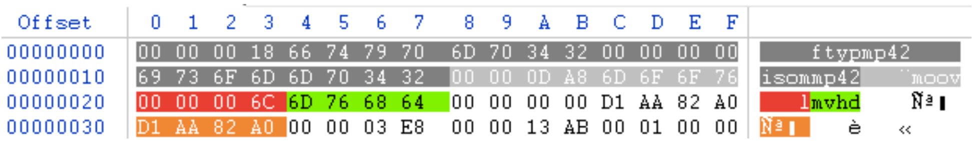
The Movie Header Box defines the characteristics of the media data contained within the file and contains a number of useful pieces of information; in this example: creation time, modification time, time scale, and duration. At an offset of 0x0C from the start of the ‘mvhd’ box is the creation time of the example file presented in a 32-bit integer in big endian that represents the number of seconds since midnight, January 1, 1904 in UTC time. This was the same timing scheme used for the Mac OS’s Hierarchical File System up through OS 9 and was also the timestamp format of the Palm OS but now this epoch time system is really only used as the encoded time in MPEG-4 and QuickTime files.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00      ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª▌
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª▌   è   «
```

Figure 10. MPEG-4 Creation Timestamp

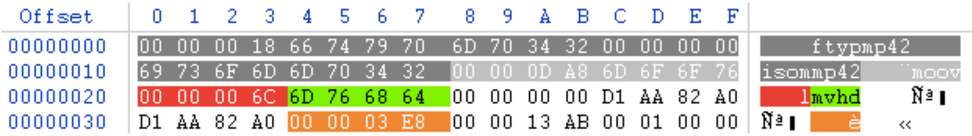
The modification time of the file is contained in the same time format as the creation time in four bytes at the offset of 0x10 from the beginning of the ‘mvhd’ box. In the case of this example file, it is identical to the creation time of the file.



```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00     ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª     è   «
```

Figure 11. MPEG-4 Modification Timestamp

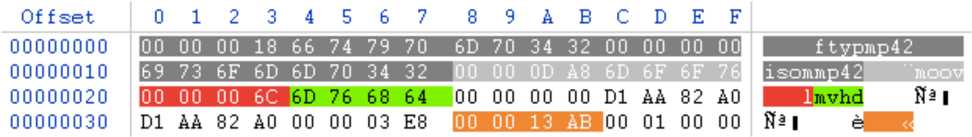
The following four bytes at offset 0x14 contain the time scale of the file presented as an integer that represents the number of time units that pass in one second. In this case, a value of 0x3E8 represents a time scale 1/1000th of a second, or one millisecond.



```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00     ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª     è   «
```

Figure 12. Movie Header Box Time Scale

At an offset of 0x18 from the start of the ‘mvhd’ box are four bytes that represent the duration of the file. In this example: 0x13AB or 5035 milliseconds. The example file has a duration of 5.035 seconds.

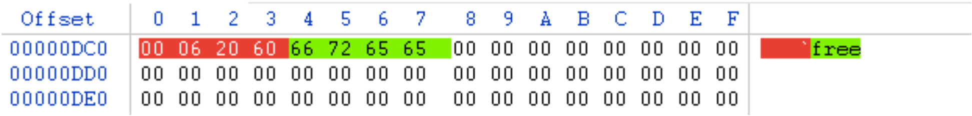


```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 18 66 74 79 70  6D 70 34 32 00 00 00 00     ftypmp42
00000010  69 73 6F 6D 6D 70 34 32  00 00 0D A8 6D 6F 6F 76  isommp42  ¨moov
00000020  00 00 00 6C 6D 76 68 64  00 00 00 00 D1 AA 82 A0     lmvhd    Ñª
00000030  D1 AA 82 A0 00 00 03 E8  00 00 13 AB 00 01 00 00  Ñª     è   «
```

Figure 13. Movie Header Box File Duration

The Free Box

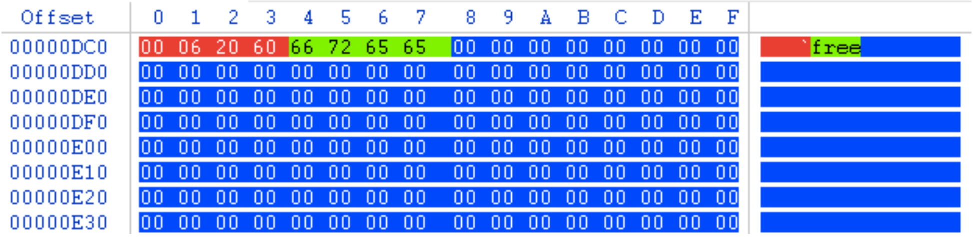
3496 bytes from the starting point of our ‘moov’ box at 0x0DA8 starts our next top-level box at offset 0x0DC0. The size of this box is 0x62060.



```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000DC0  00 06 20 60 66 72 65 65  00 00 00 00 00 00 00 00     `free
00000DD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000DE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Figure 14. Free Box Size and Type

The free box is defined by the ISO standard as being irrelevant and that its contents may be ignored[13]. In this example, the contents of the free box is filled entirely with zeroes. Throughout the files examined for this paper, there were other examples of free boxes as well as skip boxes whose contents and function are identical to the free box.

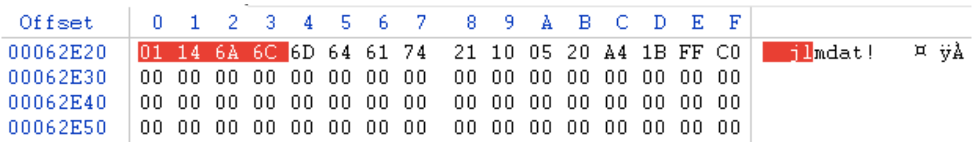


```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000DC0  00 06 20 60 66 72 65 65  00 00 00 00 00 00 00 00    `free
00000DD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000DE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000DF0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000E00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000E10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000E20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000E30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Figure 15. Free Box Contents

The Movie Data Box

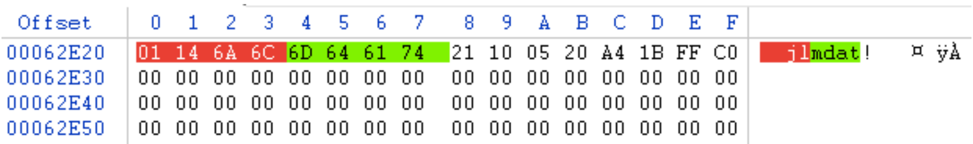
401,504 bytes from the start of the free box is our next top-level box measuring 0x1146A6C bytes. This is the final top level box in this example file and while the ISO standard would allow its size to be represented by 0x00 because its contents fill the remainder of the file, the manufacturer has chosen to define the size of the box nonetheless. In the files examined for this paper no Movie Data Box was defined as a size of 0x00.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00062E20  01 14 6A 6C 6D 64 61 74  21 10 05 20 A4 1B FF C0    jlmdat!    ¤ ÿÀ
00062E30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E40  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Figure 16. Movie Data Box Size

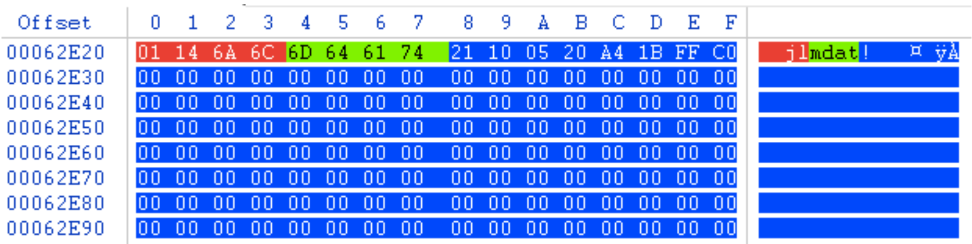
The final top-level box in this example file is ‘mdat’. The Media Data Box contains the media data of the file, in this case the compressed audio and video stream. A file may have multiple ‘mdat’ boxes containing multiple data streams or no ‘mdat’ box whatsoever if the file in question is acting only as a pointer to media data in other files.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00062E20  01 14 6A 6C 6D 64 61 74  21 10 05 20 A4 1B FF C0    jlmdat!    ¤ ÿÀ
00062E30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E40  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Figure 17. Movie Data Box Type

In this example, there is a single media data box containing a single media data stream. This was the case for all of the files examined for this paper.



```
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00062E20   01 14 6A 6C 6D 64 61 74  21 10 05 20 A4 1B FF C0    jlmdat!    ¤ ÿÀ
00062E30   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E40   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E50   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E60   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E70   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E80   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00062E90   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Figure 18. Movie Data Box Contents

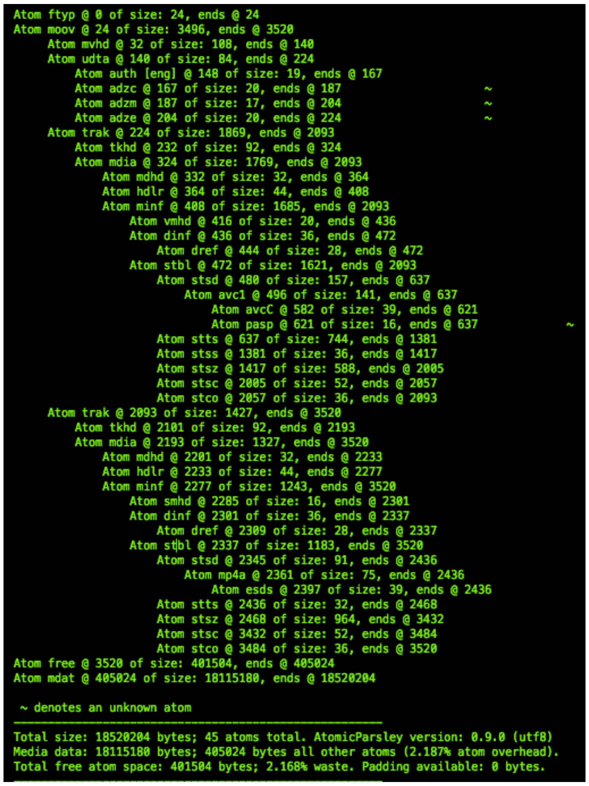
Tools for Analysis

Parsing the file structure of MPEG-4 files manually is a necessary means of understanding the box structure of a file, however, to examine a larger collection of video files, it was necessary to incorporate a number of software tools for analysis. There are a number of software tools readily available online for a variety of operating systems but two in particular were invaluable for analyzing this collection of video files. Each one focused the example file in a different way and both are freely available. The methods for using these tools should be validated in order to insure that they are reporting correct information and can be considered a forensically sound tool. It is important to note that in the research for this paper there were many instances where one tool could authenticate a file as being original to its device but by utilizing both tools many points of comparison can be identified to authenticate a given file.

AtomicParsley was used to determine container structure of the files. MediaInfo was used to interpret the contents of these containers. For the hexadecimal analysis a variety of hexadecimal editors were used including WInhex, 010 Editor, and the native Unix command 'hexdump' to carve individual boxes based on the sizes and offsets returned by AtomicParsley in order to validate the method.

AtomicParsley

AtomicParsley is a piece of software released under the terms of the GNU General Public License and available online at https://bitbucket.org/wez/atomicparsley/. Originally developed by puck_lock and currently maintained by Wez Furlong and Oleg Oshmyan, AtomicParsley will parse the box structure of a MPEG-4 file and output it to an easily readable format displaying the size and structure of the boxes.



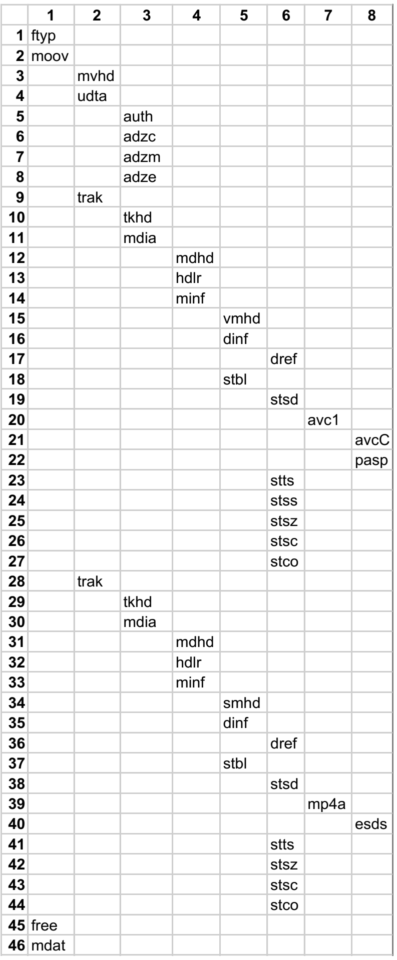
```
Atom ftyp @ 0 of size: 24, ends @ 24
Atom moov @ 24 of size: 3496, ends @ 3520
     Atom mvhd @ 32 of size: 108, ends @ 140
     Atom udta @ 140 of size: 84, ends @ 224
         Atom auth [eng] @ 148 of size: 19, ends @ 167
         Atom adzc @ 167 of size: 20, ends @ 187                      ~
         Atom adzm @ 187 of size: 17, ends @ 204                      ~
         Atom adze @ 204 of size: 20, ends @ 224                      ~
     Atom trak @ 224 of size: 1869, ends @ 2093
         Atom tkhd @ 232 of size: 92, ends @ 324
         Atom mdia @ 324 of size: 1769, ends @ 2093
             Atom mdhd @ 332 of size: 32, ends @ 364
             Atom hdlr @ 364 of size: 44, ends @ 408
             Atom minf @ 408 of size: 1685, ends @ 2093
                 Atom vmhd @ 416 of size: 20, ends @ 436
                 Atom dinf @ 436 of size: 36, ends @ 472
                     Atom dref @ 444 of size: 28, ends @ 472
                 Atom stbl @ 472 of size: 1621, ends @ 2093
                     Atom stsd @ 480 of size: 157, ends @ 637
                         Atom avc1 @ 496 of size: 141, ends @ 637
                             Atom avcC @ 582 of size: 39, ends @ 621
                             Atom pasp @ 621 of size: 16, ends @ 637              ~
                     Atom stts @ 637 of size: 744, ends @ 1381
                     Atom stss @ 1381 of size: 36, ends @ 1417
                     Atom stsz @ 1417 of size: 588, ends @ 2005
                     Atom stsc @ 2005 of size: 52, ends @ 2057
                     Atom stco @ 2057 of size: 36, ends @ 2093
     Atom trak @ 2093 of size: 1427, ends @ 3520
         Atom tkhd @ 2101 of size: 92, ends @ 2193
         Atom mdia @ 2193 of size: 1327, ends @ 3520
             Atom mdhd @ 2201 of size: 32, ends @ 2233
             Atom hdlr @ 2233 of size: 44, ends @ 2277
             Atom minf @ 2277 of size: 1243, ends @ 3520
                 Atom smhd @ 2285 of size: 16, ends @ 2301
                 Atom dinf @ 2301 of size: 36, ends @ 2337
                     Atom dref @ 2309 of size: 28, ends @ 2337
                 Atom stbl @ 2337 of size: 1183, ends @ 3520
                     Atom stsd @ 2345 of size: 91, ends @ 2436
                         Atom mp4a @ 2361 of size: 75, ends @ 2436
                             Atom esds @ 2397 of size: 39, ends @ 2436
                     Atom stts @ 2436 of size: 32, ends @ 2468
                     Atom stsz @ 2468 of size: 964, ends @ 3432
                     Atom stsc @ 3432 of size: 52, ends @ 3484
                     Atom stco @ 3484 of size: 36, ends @ 3520
Atom free @ 3520 of size: 401504, ends @ 405024
Atom mdat @ 405024 of size: 18115180, ends @ 18520204

 ~ denotes an unknown atom
------------------------------------------------------
Total size: 18520204 bytes; 45 atoms total. AtomicParsley version: 0.9.0 (utf8)
Media data: 18115180 bytes; 405024 bytes all other atoms (2.187% atom overhead).
Total free atom space: 401504 bytes; 2.168% waste. Padding available: 0 bytes.
------------------------------------------------------
```

Figure 19. AtomicParsley Example Output

In this example, the structure of our example file can quickly be identified and the nested structure of the boxes becomes clear. Manually parsing the file and comparing the results can validate the output of AtomicParsley. The size of each individual box is not important for the purpose of authentication. When recording multiple videos with the same device, variability in the size of boxes was observed, even when video files were created to be as similar as possible by matching settings and duration. However, there were no observed instances of a variability in the structure of boxes when creating multiple files using matching settings on a given device. This consistency in structure allows the examiner to create a framework to authenticate MPEG-4 video files.

It is important to note that Atomic Parsley reports boxes that are not part of its database of valid box types with a '~' and defines them as unknown atoms. These unknown atoms can be considered an excellent piece of identifying information due to the extensible nature of the MP4 standard. In the research for this paper, a number of unregistered boxes were identified, some of which contained a wealth of identifying data. The MP4 Registration Authority maintains the standards for codecs[16], file types[14], and box types[17]. By design, an unknown box will not prevent a file from being opened. By design, if an unknown box type is encountered, it will simply be ignored by the playback software.

By using the output of AtomicParsley, it is possible to create a table representative of the box structure of the example file. This will allow a visual inspection of the file structure and allow the examiner to communicate about the nature of the structure. In the case of our example, 'ftyp', 'moov', 'free', and 'mdat' are all in the 1st or top tier of the file. The 'moov' box is the only box in our file with nested containers: 'mvhd', 'udta', and two 'trak' boxes containing the video and audio streams individually. The total number of boxes can quickly be identified, in this example file there are 46 total boxes. The depth of the boxes can also be described. In this example file, there is a depth of 8 boxes. The ‘moov’ box contains ‘trak’, which contains ‘mdia’, which contains ‘minf’, which contains ‘stbl’, which contains ‘stsd’, which contains ‘avc1’, which contains ‘avcC’ and ‘pasp’. Rather than using such lengthy sentences to describe the structure of these containers, the creation of a table to visualize the file structure is invaluable when performing comparisons.



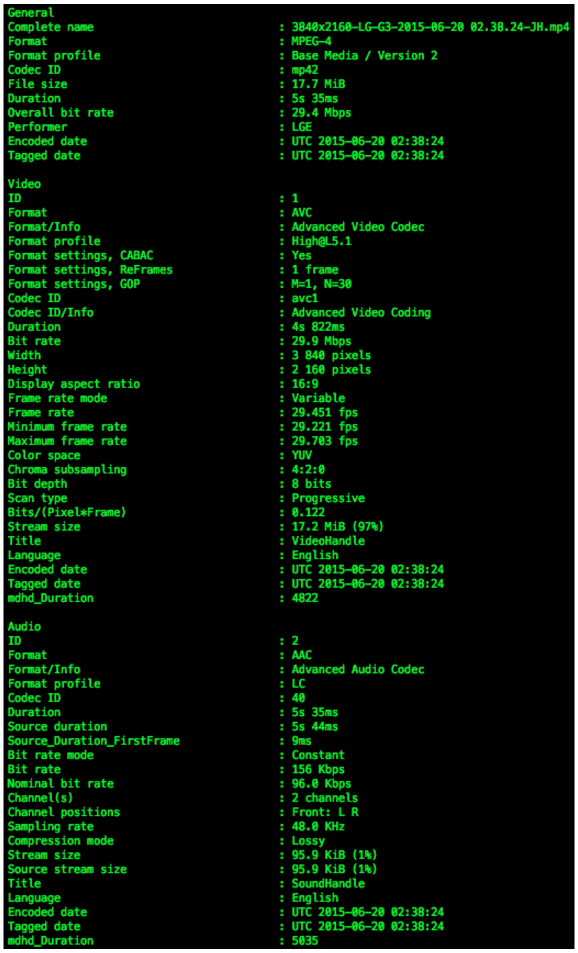
	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

Figure 20. LG G3 Structure

MediaInfo

Another valuable tool in the analysis of MPEG-4 video files is MediaInfo. Released as Open Source software under the BSD license, MediaInfo is available online at https://mediaarea.net/en/MediaInfo. For the purpose of the examinations in this paper, the CLI (Command Line Interface) version was used. MediaInfo provides a comprehensive output of the properties of a video file. MediaInfo makes no attempt to examine the structure of an input file but it excels at quickly parsing out the contents of these containers and presenting the properties of the video container, audio container, and the file itself. As a tool, MediaInfo was most useful when used to compare files from the same manufacturer that otherwise shared an identical MPEG-4 box structure.

After using MediaInfo to analyze the collection of files it became clear that as a tool it yielded certain inconsistencies when examining the properties of a file which will be described on page 23. It is imperative to understand that MediaInfo should not be relied on as the sole tool when working to authenticate a file due to these inconsistencies. A forensic examiner must understand the limitations of MediaInfo as a tool and not base any meaningful conclusions on its otherwise inconsistent results.



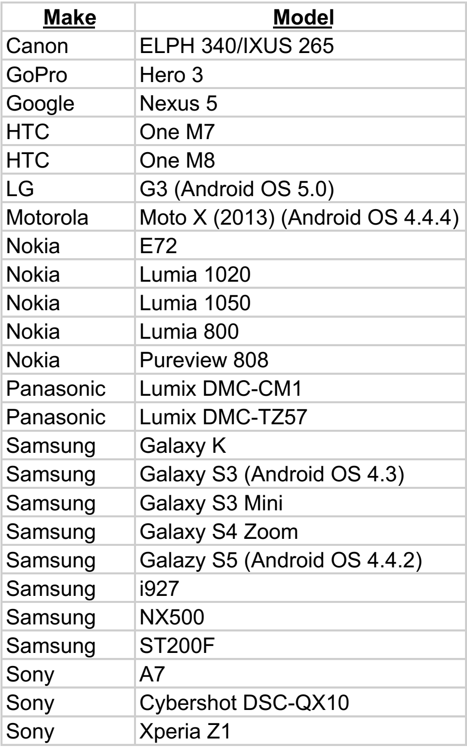
```
General
Complete name                            : 3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4
Format                                   : MPEG-4
Format profile                           : Base Media / Version 2
Codec ID                                 : mp42
File size                                : 17.7 MiB
Duration                                 : 5s 35ms
Overall bit rate                         : 29.4 Mbps
Performer                                : LGE
Encoded date                             : UTC 2015-06-20 02:38:24
Tagged date                              : UTC 2015-06-20 02:38:24

Video
ID                                       : 1
Format                                   : AVC
Format/Info                              : Advanced Video Codec
Format profile                           : High@L5.1
Format settings, CABAC                   : Yes
Format settings, ReFrames                : 1 frame
Format settings, GOP                     : M=1, N=30
Codec ID                                 : avc1
Codec ID/Info                            : Advanced Video Coding
Duration                                 : 4s 822ms
Bit rate                                 : 29.9 Mbps
Width                                    : 3 840 pixels
Height                                   : 2 160 pixels
Display aspect ratio                     : 16:9
Frame rate mode                          : Variable
Frame rate                               : 29.451 fps
Minimum frame rate                       : 29.221 fps
Maximum frame rate                       : 29.703 fps
Color space                              : YUV
Chroma subsampling                       : 4:2:0
Bit depth                                : 8 bits
Scan type                                : Progressive
Bits/(Pixel*Frame)                       : 0.122
Stream size                              : 17.2 MiB (97%)
Title                                    : VideoHandle
Language                                 : English
Encoded date                             : UTC 2015-06-20 02:38:24
Tagged date                              : UTC 2015-06-20 02:38:24
mdhd_Duration                            : 4822

Audio
ID                                       : 2
Format                                   : AAC
Format/Info                              : Advanced Audio Codec
Format profile                           : LC
Codec ID                                 : 40
Duration                                 : 5s 35ms
Source duration                          : 5s 44ms
Source_Duration_FirstFrame               : 9ms
Bit rate mode                            : Constant
Bit rate                                 : 156 Kbps
Nominal bit rate                         : 96.0 Kbps
Channel(s)                               : 2 channels
Channel positions                        : Front: L R
Sampling rate                            : 48.0 KHz
Compression mode                         : Lossy
Stream size                              : 95.9 KiB (1%)
Source stream size                       : 95.9 KiB (1%)
Title                                    : SoundHandle
Language                                 : English
Encoded date                             : UTC 2015-06-20 02:38:24
Tagged date                              : UTC 2015-06-20 02:38:24
mdhd_Duration                            : 5035
```

Figure 21. LG G3 MediaInfo Output

CHAPTER V

ANALYSIS OF CAMERA FILES

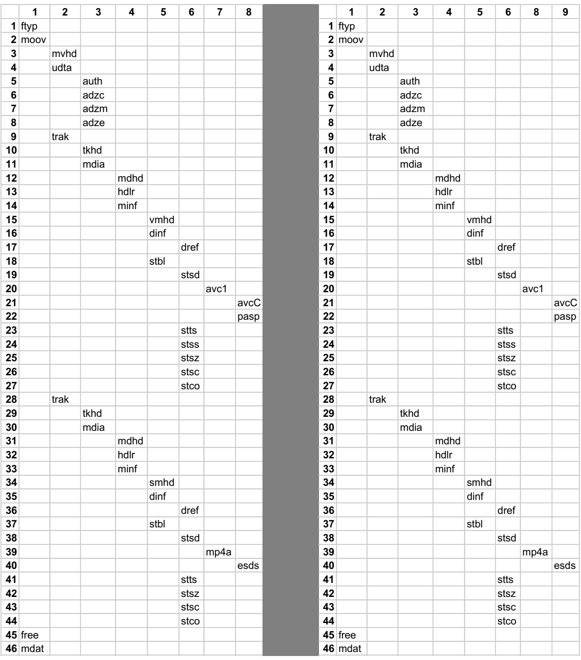
When beginning to examine the structure of the files for this paper, the extensible nature of the MPEG-4 standard became readily apparent. There are similarities in the box structure between devices and in some cases the structure is identical when comparing the structure of devices from the same manufacturer. In these cases, it is important to examine the file properties using MediaInfo as the contents of the boxes can hold important pieces of information that will aid in helping to authenticate the file to the device on which it was created. The following devices were examined for this paper:



Make	Model
Canon	ELPH 340/IXUS 265
GoPro	Hero 3
Google	Nexus 5
HTC	One M7
HTC	One M8
LG	G3 (Android OS 5.0)
Motorola	Moto X (2013) (Android OS 4.4.4)
Nokia	E72
Nokia	Lumia 1020
Nokia	Lumia 1050
Nokia	Lumia 800
Nokia	Pureview 808
Panasonic	Lumix DMC-CM1
Panasonic	Lumix DMC-TZ57
Samsung	Galaxy K
Samsung	Galaxy S3 (Android OS 4.3)
Samsung	Galaxy S3 Mini
Samsung	Galaxy S4 Zoom
Samsung	Galazy S5 (Android OS 4.4.2)
Samsung	i927
Samsung	NX500
Samsung	ST200F
Sony	A7
Sony	Cybershot DSC-QX10
Sony	Xperia Z1

Figure 22. List of Devices Analyzed for this Paper

To begin, two video clips were created using the LG G3 in its full resolution mode. In order to validate the method of using AtomicParsley as a tool and the LG G3's ability to produce repeatable results in file structure, both files were analyzed and compared.

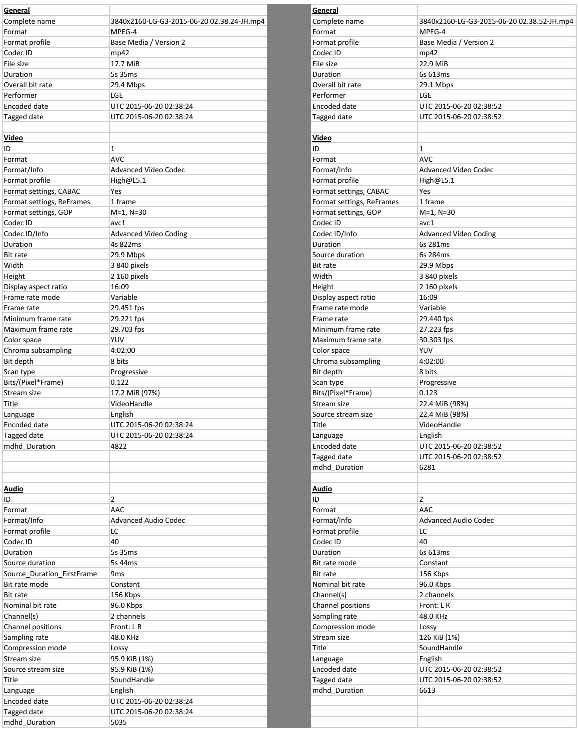


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

Figure 23. Comparison of two LG G3 Samples to Validate Structure

The two video clips show a matching structure of MPEG-4 box containers and it is now necessary to validate the method of using our second software tool MediaInfo. For this validation, the properties of the same two video files were compared.



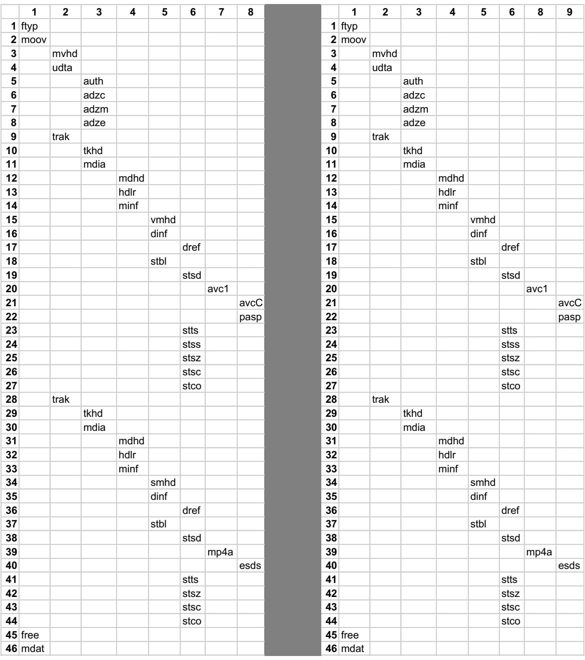
General			General	
Complete name	3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4		Complete name	3840x2160-LG-G3-2015-06-20 02.38.52-JH.mp4
Format	MPEG-4		Format	MPEG-4
Format profile	Base Media / Version 2		Format profile	Base Media / Version 2
Codec ID	mp42		Codec ID	mp42
File size	17.7 MiB		File size	22.9 MiB
Duration	5s 35ms		Duration	6s 613ms
Overall bit rate	29.4 Mbps		Overall bit rate	29.1 Mbps
Performer	LGE		Performer	LGE
Encoded date	UTC 2015-06-20 02:38:24		Encoded date	UTC 2015-06-20 02:38:52
Tagged date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-06-20 02:38:52
Video			Video	
ID	1		ID	1
Format	AVC		Format	AVC
Format/Info	Advanced Video Codec		Format/Info	Advanced Video Codec
Format profile	High@L5.1		Format profile	High@L5.1
Format settings, CABAC	Yes		Format settings, CABAC	Yes
Format settings, ReFrames	1 frame		Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30		Format settings, GOP	M=1, N=30
Codec ID	avc1		Codec ID	avc1
Codec ID/Info	Advanced Video Coding		Codec ID/Info	Advanced Video Coding
Duration	4s 822ms		Duration	6s 281ms
Bit rate	29.9 Mbps		Source duration	6s 284ms
Width	3 840 pixels		Bit rate	29.9 Mbps
Height	2 160 pixels		Width	3 840 pixels
Display aspect ratio	16:09		Height	2 160 pixels
Frame rate mode	Variable		Display aspect ratio	16:09
Frame rate	29.451 fps		Frame rate mode	Variable
Minimum frame rate	29.221 fps		Frame rate	29.440 fps
Maximum frame rate	29.703 fps		Minimum frame rate	27.223 fps
Color space	YUV		Maximum frame rate	30.303 fps
Chroma subsampling	4:02:00		Color space	YUV
Bit depth	8 bits		Chroma subsampling	4:02:00
Scan type	Progressive		Bit depth	8 bits
Bits/(Pixel*Frame)	0.122		Scan type	Progressive
Stream size	17.2 MiB (97%)		Bits/(Pixel*Frame)	0.123
Title	VideoHandle		Stream size	22.4 MiB (98%)
Language	English		Source stream size	22.4 MiB (98%)
Encoded date	UTC 2015-06-20 02:38:24		Title	VideoHandle
Tagged date	UTC 2015-06-20 02:38:24		Language	English
mdhd_Duration	4822		Encoded date	UTC 2015-06-20 02:38:52
			Tagged date	UTC 2015-06-20 02:38:52
			mdhd_Duration	6281
Audio			Audio	
ID	2		ID	2
Format	AAC		Format	AAC
Format/Info	Advanced Audio Codec		Format/Info	Advanced Audio Codec
Format profile	LC		Format profile	LC
Codec ID	40		Codec ID	40
Duration	5s 35ms		Duration	6s 613ms
Source duration	5s 44ms		Bit rate mode	Constant
Source_Duration_FirstFrame	9ms		Bit rate	156 Kbps
Bit rate mode	Constant		Nominal bit rate	96.0 Kbps
Bit rate	156 Kbps		Channel(s)	2 channels
Nominal bit rate	96.0 Kbps		Channel positions	Front: L R
Channel(s)	2 channels		Sampling rate	48.0 KHz
Channel positions	Front: L R		Compression mode	Lossy
Sampling rate	48.0 KHz		Stream size	126 KiB (1%)
Compression mode	Lossy		Title	SoundHandle
Stream size	95.9 KiB (1%)		Language	English
Source stream size	95.9 KiB (1%)		Encoded date	UTC 2015-06-20 02:38:52
Title	SoundHandle		Tagged date	UTC 2015-06-20 02:38:52
Language	English		mdhd_Duration	6613
Encoded date	UTC 2015-06-20 02:38:24			
Tagged date	UTC 2015-06-20 02:38:24			
mdhd_Duration	5035			

Figure 24. Comparison of two LG G3 Samples to Validate MediaInfo Properties

When comparing the two files, MediaInfo reported a property in one file that it didn't in the other: Source Duration. A series of additional test videos were created originally thinking that the presence of the Source Duration property might correlate to the duration of the video itself, in other words, a short video would not store that property but a longer video would. In testing, no correlation could be found to explain the presence or absence of this property reporting in MediaInfo. However, the box structure analysis with AtomicParsley did remain consistent throughout testing. In this case, the presence or absence of the Source Duration property has no effect on the authentication of the LG G3 video clips being examined but it is important to make note of any inconsistencies when examining files.

The Source Duration property was attached to both the audio and video tracks so the Track Box ('trak') and Media Header Box ('mdia') for each stream were parsed manually and each contained duration information. This is an excellent demonstration of the importance that should be placed on parsing manually when any inconsistencies are observed, in order to better understand the output of the tools being used for analysis and to better understand the structure of the files in question before making a meaningful decision based on the results of analysis.

To continue validating the LG G3, one of the full resolution video clips was compared to a lower resolution, slow motion recording mode available on the device. The structure of these two files were then parsed and compared.

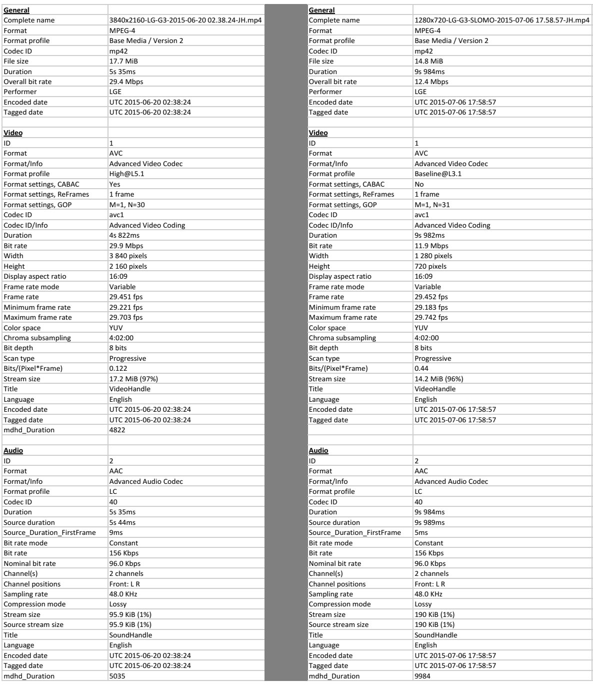


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

Figure 25. Comparison of two LG G3 File Structures in Different Recording Modes (Full Resolution vs. Slow Motion)

The box structure using the two different modes on the LG G3 remained consistent. For the sake of further validation, the files were compared using Media Info.



General			**General**	
Complete name	3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4		Complete name	1280x720-LG-G3-SLOMO-2015-07-06 17.58.57-JH.mp4
Format	MPEG-4		Format	MPEG-4
Format profile	Base Media / Version 2		Format profile	Base Media / Version 2
Codec ID	mp42		Codec ID	mp42
File size	17.7 MiB		File size	14.8 MiB
Duration	5s 35ms		Duration	9s 984ms
Overall bit rate	29.4 Mbps		Overall bit rate	12.4 Mbps
Performer	LGE		Performer	LGE
Encoded date	UTC 2015-06-20 02:38:24		Encoded date	UTC 2015-07-06 17:58:57
Tagged date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-07-06 17:58:57
Video			**Video**	
ID	1		ID	1
Format	AVC		Format	AVC
Format/Info	Advanced Video Codec		Format/Info	Advanced Video Codec
Format profile	High@L5.1		Format profile	Baseline@L3.1
Format settings, CABAC	Yes		Format settings, CABAC	No
Format settings, ReFrames	1 frame		Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30		Format settings, GOP	M=1, N=31
Codec ID	avc1		Codec ID	avc1
Codec ID/Info	Advanced Video Coding		Codec ID/Info	Advanced Video Coding
Duration	4s 822ms		Duration	9s 982ms
Bit rate	29.9 Mbps		Bit rate	11.9 Mbps
Width	3 840 pixels		Width	1 280 pixels
Height	2 160 pixels		Height	720 pixels
Display aspect ratio	16:09		Display aspect ratio	16:09
Frame rate mode	Variable		Frame rate mode	Variable
Frame rate	29.451 fps		Frame rate	29.452 fps
Minimum frame rate	29.221 fps		Minimum frame rate	29.183 fps
Maximum frame rate	29.703 fps		Maximum frame rate	29.742 fps
Color space	YUV		Color space	YUV
Chroma subsampling	4:02:00		Chroma subsampling	4:02:00
Bit depth	8 bits		Bit depth	8 bits
Scan type	Progressive		Scan type	Progressive
Bits/(Pixel*Frame)	0.122		Bits/(Pixel*Frame)	0.44
Stream size	17.2 MiB (97%)		Stream size	14.2 MiB (96%)
Title	VideoHandle		Title	VideoHandle
Language	English		Language	English
Encoded date	UTC 2015-06-20 02:38:24		Encoded date	UTC 2015-07-06 17:58:57
Tagged date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-07-06 17:58:57
mdhd_Duration	4822			
Audio			**Audio**	
ID	2		ID	2
Format	AAC		Format	AAC
Format/Info	Advanced Audio Codec		Format/Info	Advanced Audio Codec
Format profile	LC		Format profile	LC
Codec ID	40		Codec ID	40
Duration	5s 35ms		Duration	9s 984ms
Source duration	5s 44ms		Source duration	9s 989ms
Source_Duration_FirstFrame	9ms		Source_Duration_FirstFrame	5ms
Bit rate mode	Constant		Bit rate mode	Constant
Bit rate	156 Kbps		Bit rate	156 Kbps
Nominal bit rate	96.0 Kbps		Nominal bit rate	96.0 Kbps
Channel(s)	2 channels		Channel(s)	2 channels
Channel positions	Front: L R		Channel positions	Front: L R
Sampling rate	48.0 KHz		Sampling rate	48.0 KHz
Compression mode	Lossy		Compression mode	Lossy
Stream size	95.9 KiB (1%)		Stream size	190 KiB (1%)
Source stream size	95.9 KiB (1%)		Source stream size	190 KiB (1%)
Title	SoundHandle		Title	SoundHandle
Language	English		Language	English
Encoded date	UTC 2015-06-20 02:38:24		Encoded date	UTC 2015-07-06 17:58:57
Tagged date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-07-06 17:58:57
mdhd_Duration	5035		mdhd_Duration	9984

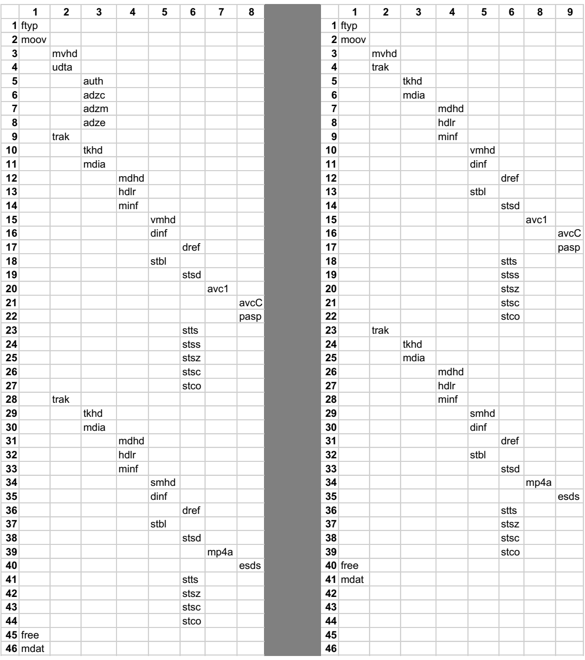
Figure 26. Comparison of two LG G3 File Properties in Different Recording Modes (Full Resolution vs. Slow Motion)

The results reported by MediaInfo confirmed the different properties of the two files but again reported some properties in one file and not in the other. In this case, the Media Header Box (‘mdhd’) duration was not reported in the lower resolution file. Again, this information exists in both files but MediaInfo failed to report it for the second file. Further analysis of files using MediaInfo revealed that the absence or presence in reporting Source Duration or Media Header Box (‘mdhd’) duration occurred throughout the analysis for this paper. Multiple tests of multiple files were performed and in some cases the same file was examined multiple times. MediaInfo never returned a different result when examining the same file multiple times but there were simply some files that it would report these properties on and others that it would not.

After establishing that the LG G3 creates files with consistent structure, a comparison was made with the Motorola Moto X 2013. The Motorola Moto X

2013 would only record in one mode; the device was validated against itself to confirm that it made consistently structured recordings.

By visualizing the structure of these two files, it is possible to quickly compare them in order to determine if they have a matching structure of boxes or if they are different in some way. In the case of the LG G3 and the Motorola Moto X 2013, the file structures are very similar but the LG G3 includes a User Data ('udta') box which contains a number of boxes that are unique to the LG device: 'auth', 'adzc', 'adzm', and 'adze'. The ISO/IEC 14496-12:2005(E) standard only defines a copyright notice to be contained inside a User Data Box ('udta') but it is an extensible container which can be used as the manufacturer sees fit as in the case of the LG G3. Were it not for this 'udta' box and its contents, the structure of the two files is otherwise identical and it would be necessary to parse out the identifying properties of the files themselves.

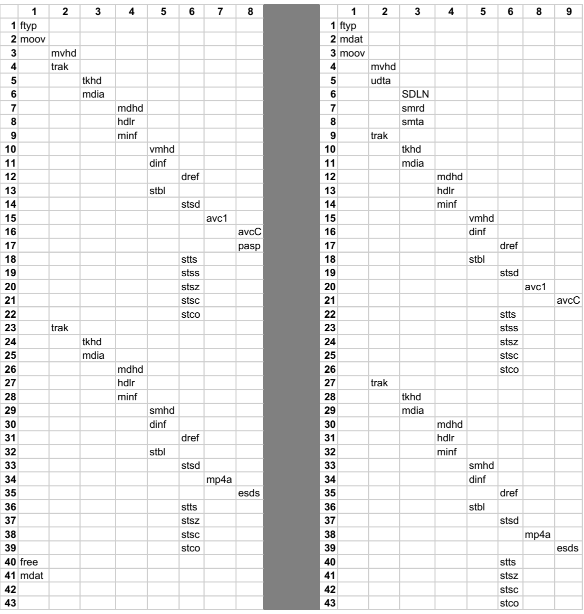


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	moov							
3		mvhd						
4		trak						
5			tkhd					
6			mdia					
7				mdhd				
8				hdlr				
9				minf				
10					vmhd			
11					dinf			
12						dref		
13					stbl			
14						stsd		
15							avc1	
16								avcC
17								pasp
18						stts		
19						stss		
20						stsz		
21						stsc		
22						stco		
23		trak						
24			tkhd					
25			mdia					
26				mdhd				
27				hdlr				
28				minf				
29					smhd			
30					dinf			
31						dref		
32					stbl			
33						stsd		
34							mp4a	
35								esds
36						stts		
37						stsz		
38						stsc		
39						stco		
40	free							
41	mdat							
42								
43								
44								
45								
46								

Figure 27. Comparison of LG G3 and Moto X (2013) Structure

When comparing the Motorola Moto X and the Samsung S5, the structure is clearly unique between the two devices. Most notably, the Samsung S5 places the ‘moov’ box after the ‘mdat’ box but Samsung also inserts a User Data (‘udta’) box containing three additional boxes: ‘SDLN’, ‘smrd’, and ‘smta’. The placement of the Movie Data Box (‘mdat’) before the Movie Box (‘moov’) is notable because ISO/IEC 14496-12:2005(E) specifically recommends placing the descriptive information of a MPEG-4 file before the data itself. This recommendation is to facilitate the streaming of the video. In this case, the video from the Moto X could be streamed because the file type header and descriptive data for the video content itself would be received then the playback would begin streaming the audio and video data contained in the ‘mdat’ box. The file created by the Samsung Galaxy S5 could not be streamed because in order for playback to occur, the entire file would need to be loaded in order to receive the descriptive content in the ‘moov’ box to then be able to interpret the data contained in the ‘mdat’ box.

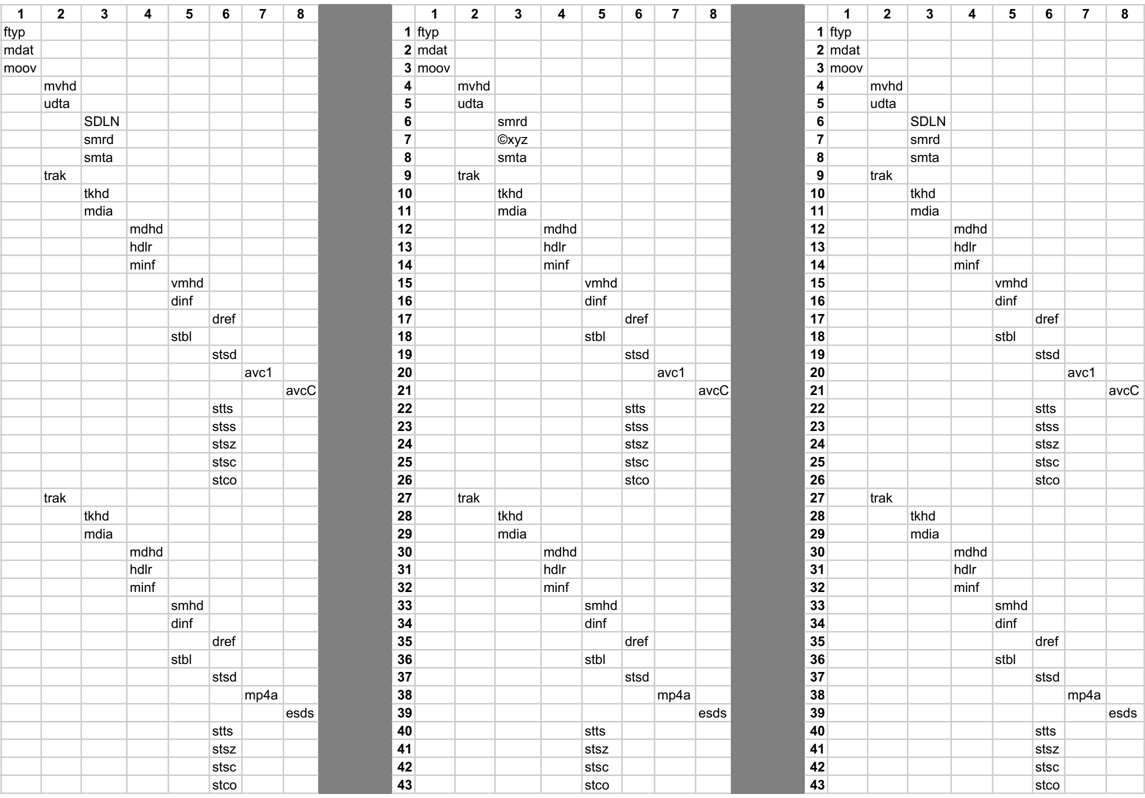


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		trak						
5			tkhd					
6			mdia					
7				mdhd				
8				hdlr				
9				minf				
10					vmhd			
11					dinf			
12						dref		
13					stbl			
14						stsd		
15							avc1	
16								avcC
17								pasp
18						stts		
19						stss		
20						stsz		
21						stsc		
22						stco		
23		trak						
24			tkhd					
25			mdia					
26				mdhd				
27				hdlr				
28				minf				
29					smhd			
30					dinf			
31						dref		
32					stbl			
33						stsd		
34							mp4a	
35								esds
36						stts		
37						stsz		
38						stsc		
39						stco		
40	free							
41	mdat							
42								
43								

	1	2	3	4	5	6	8	9
1	ftyp							
2	mdat							
3	moov							
4		mvhd						
5		udta						
6			SDLN					
7			smrd					
8			smta					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22						stts		
23						stss		
24						stsz		
25						stsc		
26						stco		
27		trak						
28			tkhd					
29			mdia					
30				mdhd				
31				hdlr				
32				minf				
33					smhd			
34					dinf			
35						dref		
36					stbl			
37						stsd		
38							mp4a	
39								esds
40						stts		
41						stsz		
42						stsc		
43						stco		

Figure 28. Comparison of Moto X and Samsung S5 Structure

When comparing file structure across Samsung devices, they are expectedly similar. The Galaxy S3 and Galaxy S5 have identical structures while the S4 Zoom has a structure that differs only slightly from the S3 and S5 in its User Data Box (‘udta’).



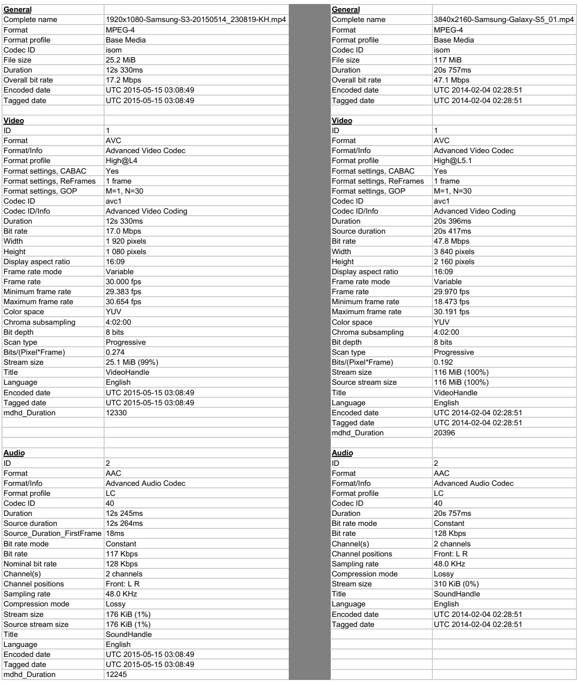
1	2	3	4	5	6	7	8
ftyp							
mdat							
moov							
	mvhd						
	udta						
		SDLN					
		smrd					
		smta					
	trak						
		tkhd					
		mdia					
			mdhd				
			hdlr				
			minf				
				vmhd			
				dinf			
					dref		
				stbl			
					stsd		
						avc1	
							avcC
					stts		
					stss		
					stsz		
					stsc		
					stco		
	trak						
		tkhd					
		mdia					
			mdhd				
			hdlr				
			minf				
				smhd			
				dinf			
					dref		
				stbl			
					stsd		
						mp4a	
							esds
					stts		
					stsz		
					stsc		
					stco		

	1	2	3	4	5	6	7	8
1	ftyp							
2	mdat							
3	moov							
4		mvhd						
5		udta						
6			smrd					
7			©xyz					
8			smta					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22						stts		
23						stss		
24						stsz		
25						stsc		
26						stco		
27		trak						
28			tkhd					
29			mdia					
30				mdhd				
31				hdlr				
32				minf				
33					smhd			
34					dinf			
35						dref		
36					stbl			
37						stsd		
38							mp4a	
39								esds
40						stts		
41						stsz		
42						stsc		
43						stco		

	1	2	3	4	5	6	7	8
1	ftyp							
2	mdat							
3	moov							
4		mvhd						
5		udta						
6			SDLN					
7			smrd					
8			smta					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22						stts		
23						stss		
24						stsz		
25						stsc		
26						stco		
27		trak						
28			tkhd					
29			mdia					
30				mdhd				
31				hdlr				
32				minf				
33					smhd			
34					dinf			
35						dref		
36					stbl			
37						stsd		
38							mp4a	
39								esds
40						stts		
41						stsz		
42						stsc		
43						stco		

Figure 29. Comparison of Samsung S3, S4 Zoom, and S5 Structure

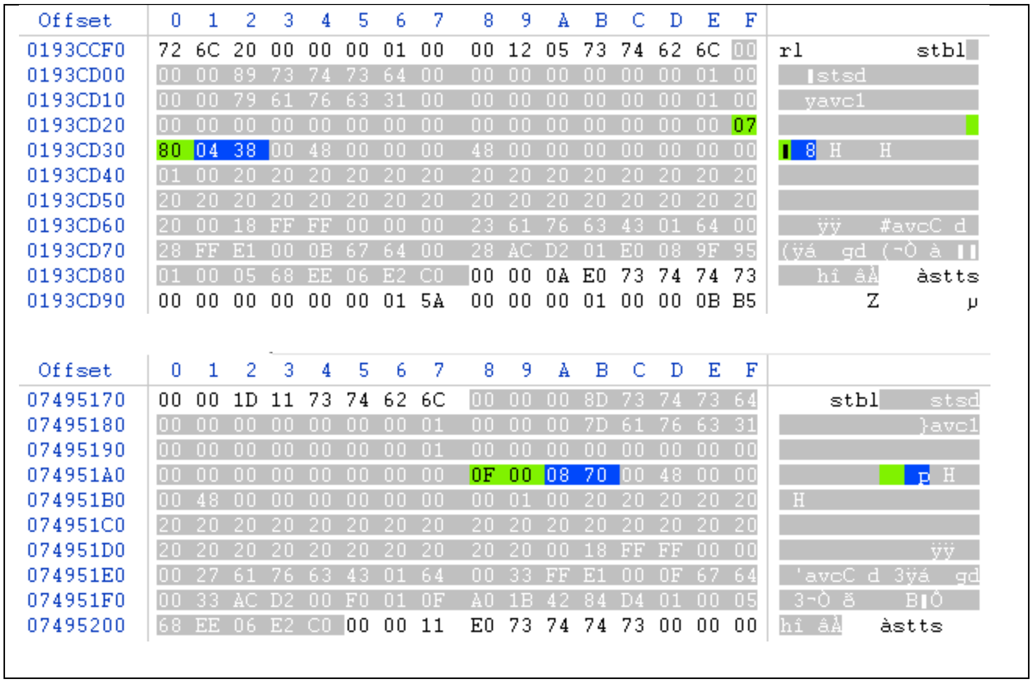
Presented with two files of identical box structure, the next step in authenticating these files should be to examine their properties in order to make further attempt to authenticate them to a known device. Using MediaInfo, the properties of these two files can be examined and compared to quickly identify any characteristics that would differentiate the two files. In the case of these two files being examined, MediaInfo reports that the resolution of the two files is different.



General		**General**	
Complete name	1920x1080-Samsung-S3-20150514_230819-KH.mp4	Complete name	3840x2160-Samsung-Galaxy-S5_01.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	Base Media	Format profile	Base Media
Codec ID	isom	Codec ID	isom
File size	25.2 MiB	File size	117 MiB
Duration	12s 330ms	Duration	20s 757ms
Overall bit rate	17.2 Mbps	Overall bit rate	47.1 Mbps
Encoded date	UTC 2015-05-15 03:08:49	Encoded date	UTC 2014-02-04 02:28:51
Tagged date	UTC 2015-05-15 03:08:49	Tagged date	UTC 2014-02-04 02:28:51
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	High@L4	Format profile	High@L5.1
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30	Format settings, GOP	M=1, N=30
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	12s 330ms	Duration	20s 396ms
Bit rate	17.0 Mbps	Source duration	20s 417ms
Width	1 920 pixels	Bit rate	47.8 Mbps
Height	1 080 pixels	Width	3 840 pixels
Display aspect ratio	16:09	Height	2 160 pixels
Frame rate mode	Variable	Display aspect ratio	16:09
Frame rate	30.000 fps	Frame rate mode	Variable
Minimum frame rate	29.383 fps	Frame rate	29.970 fps
Maximum frame rate	30.654 fps	Minimum frame rate	18.473 fps
Color space	YUV	Maximum frame rate	30.191 fps
Chroma subsampling	4:02:00	Color space	YUV
Bit depth	8 bits	Chroma subsampling	4:02:00
Scan type	Progressive	Bit depth	8 bits
Bits/(Pixel*Frame)	0.274	Scan type	Progressive
Stream size	25.1 MiB (99%)	Bits/(Pixel*Frame)	0.192
Title	VideoHandle	Stream size	116 MiB (100%)
Language	English	Source stream size	116 MiB (100%)
Encoded date	UTC 2015-05-15 03:08:49	Title	VideoHandle
Tagged date	UTC 2015-05-15 03:08:49	Language	English
mdhd_Duration	12330	Encoded date	UTC 2014-02-04 02:28:51
		Tagged date	UTC 2014-02-04 02:28:51
		mdhd_Duration	20396
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	12s 245ms	Duration	20s 757ms
Source duration	12s 264ms	Bit rate mode	Constant
Source_Duration_FirstFrame	18ms	Bit rate	128 Kbps
Bit rate mode	Constant	Channel(s)	2 channels
Bit rate	117 Kbps	Channel positions	Front: L R
Nominal bit rate	128 Kbps	Sampling rate	48.0 KHz
Channel(s)	2 channels	Compression mode	Lossy
Channel positions	Front: L R	Stream size	310 KiB (0%)
Sampling rate	48.0 KHz	Title	SoundHandle
Compression mode	Lossy	Language	English
Stream size	176 KiB (1%)	Encoded date	UTC 2014-02-04 02:28:51
Source stream size	176 KiB (1%)	Tagged date	UTC 2014-02-04 02:28:51
Title	SoundHandle		
Language	English		
Encoded date	UTC 2015-05-15 03:08:49		
Tagged date	UTC 2015-05-15 03:08:49		
mdhd_Duration	12245		

Figure 30. MediaInfo Comparison of Samsung S3 and Samsung S5

When examining the individual files, it is important to understand where MediaInfo is deriving this information. ISO/IEC 14496-12:2005(E) requires that the horizontal and vertical resolution of a file be defined in the Sample Description Box (‘stsd’) which is contained in the Sample Table Box (‘stbl’), which is ultimately contained in the Track Box (‘trak’) for the video stream of the respective files. In the Samsung Galaxy S3 and Samsung Galaxy S5, this data is represented in two unsigned 16-bit integers beginning at an offset of 0x31 from the beginning of the Sample Table Box (‘stbl’). The first two bytes represent the horizontal resolution (in green) and the second two bytes represent the vertical resolution (in blue).



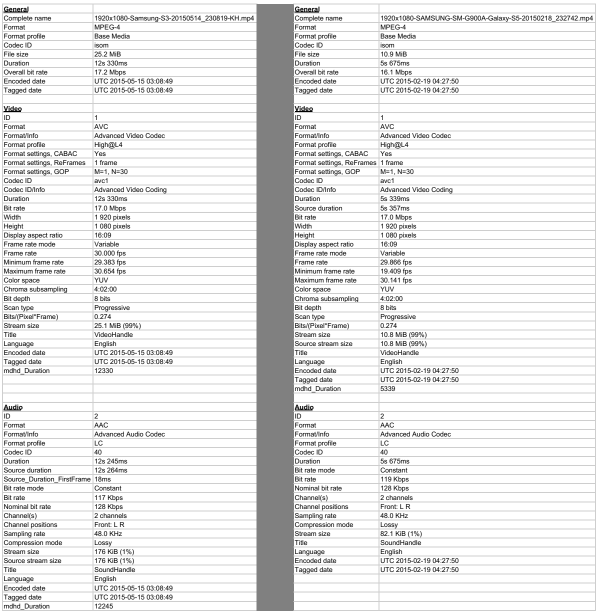
```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0193CCF0  72 6C 20 00 00 00 01 00  00 12 05 73 74 62 6C 00   rl          stbl
0193CD00  00 00 89 73 74 73 64 00  00 00 00 00 00 00 01 00     stsd
0193CD10  00 00 79 61 76 63 31 00  00 00 00 00 00 00 01 00     yavc1
0193CD20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 07
0193CD30  80 04 38 00 48 00 00 00  48 00 00 00 00 00 00 00    8 H   H
0193CD40  01 00 20 20 20 20 20 20  20 20 20 20 20 20 20 20
0193CD50  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
0193CD60  20 00 18 FF FF 00 00 00  23 61 76 63 43 01 64 00     ÿÿ   #avcC d
0193CD70  28 FF E1 00 0B 67 64 00  28 AC D2 01 E0 08 9F 95   (ÿá  gd (¬Ò à
0193CD80  01 00 05 68 EE 06 E2 C0  00 00 0A E0 73 74 74 73     hî âÀ    àstts
0193CD90  00 00 00 00 00 00 01 5A  00 00 00 01 00 00 0B B5          Z     µ
```

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
07495170  00 00 1D 11 73 74 62 6C  00 00 00 8D 73 74 73 64       stbl    stsd
07495180  00 00 00 00 00 00 00 01  00 00 00 7D 61 76 63 31              }avc1
07495190  00 00 00 00 00 00 00 01  00 00 00 00 00 00 00 00
074951A0  00 00 00 00 00 00 00 00  0F 00 08 70 00 48 00 00              p H
074951B0  00 48 00 00 00 00 00 00  00 01 00 20 20 20 20 20    H
074951C0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
074951D0  20 20 20 20 20 20 20 20  20 20 00 18 FF FF 00 00               ÿÿ
074951E0  00 27 61 76 63 43 01 64  00 33 FF E1 00 0F 67 64    'avcC d 3ÿá  gd
074951F0  00 33 AC D2 00 F0 01 0F  A0 1B 42 84 D4 01 00 05    3¬Ò ð     B Ô
07495200  68 EE 06 E2 C0 00 00 11  E0 73 74 74 73 00 00 00   hî âÀ   àstts
```

Figure 31. Comparison of ‘stbl’ Boxes in Samsung S3 (top) and S5 (bottom)

The maximum resolution that the Galaxy S3 can record is 1920x1080 where the maximum video resolution of the Galaxy S5 is 3840x2160. Therefore, in this example, while the box structure of the two files is identical, an analysis of the contents of the Sample Description Box (‘stsd’) can be examined to determine more specific properties of the video files in order to authenticate them. This is a valid means of authenticating a video whose MPEG-4 box structure is identical to determine if it is the correct resolution for the device in question. This specific technique has a limitation if a device capable of recording in a lower resolution than its maximum resolution is compared against a second device recording at the same resolution. In the study for this paper, when a Samsung Galaxy S3 recording at its maximum resolution of 1920x1080 is compared against a Samsung Galaxy S5 recording at a lower than maximum resolution of 1920x1080, the files appear identical both in structure and in metadata. MediaInfo confirms the resolutions of both files as being identical and other than small variances in the frame rate, which should not be considered a viable means of differentiating the files in this case, there is no meaningful data to exclude these two files from being a match as the same device.

This result was not unexpected or surprising. The Samsung devices show a great number of similarities in their file structure and metadata including the contents of their User Data Box (‘udta’). In this example, both devices report the same video format profile. In both Samsung files, the video format profile is reported as ‘High@L4’. Looking back at the MediaInfo output of a Samsung Galaxy S5 video recorded at 3840x2160, the video format profile is reported as ‘High@L5.1’. This is a second way to differentiate between the Samsung Galaxy S3 and Galaxy S5 recording at their maximum resolutions. These descriptors do not appear to be standardized in any way and appear to define the quality of encoding on the device.[18]

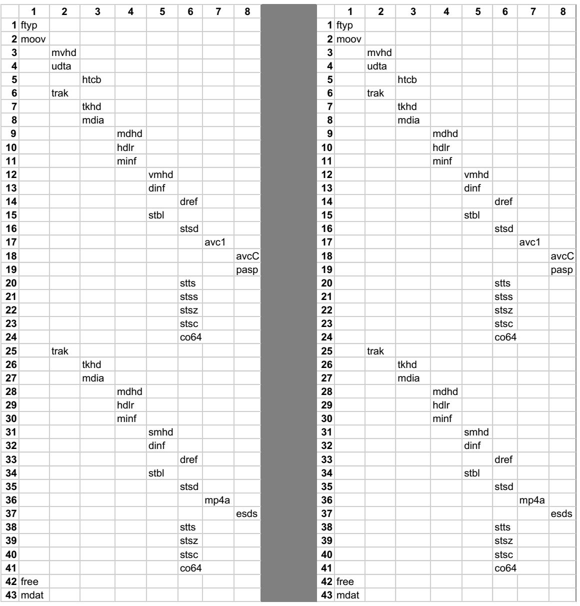


General	
Complete name	1920x1080-Samsung-S3-20150514_230819-KH.mp4
Format	MPEG-4
Format profile	Base Media
Codec ID	isom
File size	25.2 MiB
Duration	12s 330ms
Overall bit rate	17.2 Mbps
Encoded date	UTC 2015-05-15 03:08:49
Tagged date	UTC 2015-05-15 03:08:49
Video	
ID	1
Format	AVC
Format/Info	Advanced Video Codec
Format profile	High@L4
Format settings, CABAC	Yes
Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30
Codec ID	avc1
Codec ID/Info	Advanced Video Coding
Duration	12s 330ms
Bit rate	17.0 Mbps
Width	1 920 pixels
Height	1 080 pixels
Display aspect ratio	16:09
Frame rate mode	Variable
Frame rate	30.000 fps
Minimum frame rate	29.383 fps
Maximum frame rate	30.654 fps
Color space	YUV
Chroma subsampling	4:02:00
Bit depth	8 bits
Scan type	Progressive
Bits/(Pixel*Frame)	0.274
Stream size	25.1 MiB (99%)
Title	VideoHandle
Language	English
Encoded date	UTC 2015-05-15 03:08:49
Tagged date	UTC 2015-05-15 03:08:49
mdhd_Duration	12330
Audio	
ID	2
Format	AAC
Format/Info	Advanced Audio Codec
Format profile	LC
Codec ID	40
Duration	12s 245ms
Source duration	12s 264ms
Source_Duration_FirstFrame	18ms
Bit rate mode	Constant
Bit rate	117 Kbps
Nominal bit rate	128 Kbps
Channel(s)	2 channels
Channel positions	Front: L R
Sampling rate	48.0 KHz
Compression mode	Lossy
Stream size	176 KiB (1%)
Source stream size	176 KiB (1%)
Title	SoundHandle
Language	English
Encoded date	UTC 2015-05-15 03:08:49
Tagged date	UTC 2015-05-15 03:08:49
mdhd_Duration	12245

General	
Complete name	1920x1080-SAMSUNG-SM-G900A-Galaxy-S5-20150218_232742.mp4
Format	MPEG-4
Format profile	Base Media
Codec ID	isom
File size	10.9 MiB
Duration	5s 675ms
Overall bit rate	16.1 Mbps
Encoded date	UTC 2015-02-19 04:27:50
Tagged date	UTC 2015-02-19 04:27:50
Video	
ID	1
Format	AVC
Format/Info	Advanced Video Codec
Format profile	High@L4
Format settings, CABAC	Yes
Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30
Codec ID	avc1
Codec ID/Info	Advanced Video Coding
Duration	5s 339ms
Source duration	5s 357ms
Bit rate	17.0 Mbps
Width	1 920 pixels
Height	1 080 pixels
Display aspect ratio	16:09
Frame rate mode	Variable
Frame rate	29.866 fps
Minimum frame rate	19.409 fps
Maximum frame rate	30.141 fps
Color space	YUV
Chroma subsampling	4:02:00
Bit depth	8 bits
Scan type	Progressive
Bits/(Pixel*Frame)	0.274
Stream size	10.8 MiB (99%)
Source stream size	10.8 MiB (99%)
Title	VideoHandle
Language	English
Encoded date	UTC 2015-02-19 04:27:50
Tagged date	UTC 2015-02-19 04:27:50
mdhd_Duration	5339
Audio	
ID	2
Format	AAC
Format/Info	Advanced Audio Codec
Format profile	LC
Codec ID	40
Duration	5s 675ms
Bit rate mode	Constant
Bit rate	119 Kbps
Nominal bit rate	128 Kbps
Channel(s)	2 channels
Channel positions	Front: L R
Sampling rate	48.0 KHz
Compression mode	Lossy
Stream size	82.1 KiB (1%)
Title	SoundHandle
Language	English
Encoded date	UTC 2015-02-19 04:27:50
Tagged date	UTC 2015-02-19 04:27:50

Figure 32. MediaInfo Comparison of Samsung S5 Between Recording Modes

While Samsung maintains a constant structure of video format profiles across the Samsung Galaxy S3 and Galaxy S5, this is a matter left up to the manufacturer and is in no way defined by ISO/IEC 14496-12:2005(E). When applying the same technique of analysis to a different set of identically structured files from a different manufacturer, the results are different. The HTC One M7 and the HTC One M8 create files of identical MPEG-4 box structure.



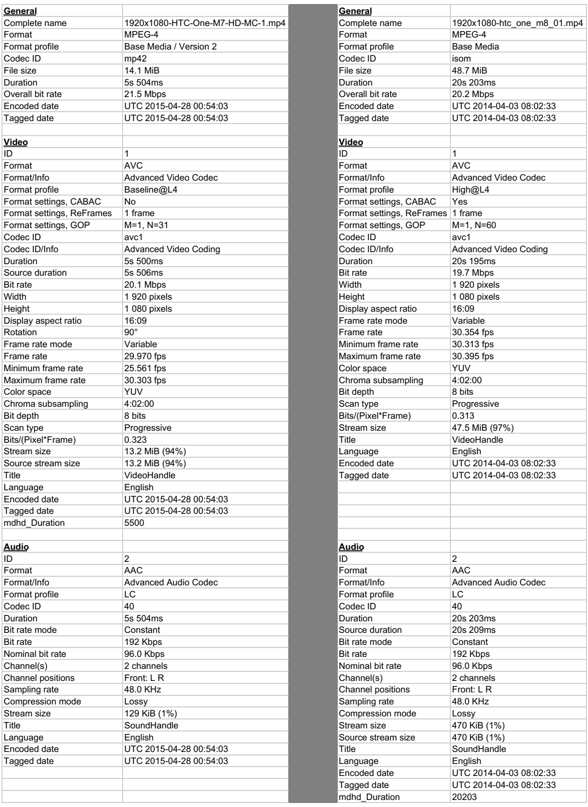
	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			htcb					
6		trak						
7			tkhd					
8			mdia					
9				mdhd				
10				hdlr				
11				minf				
12					vmhd			
13					dinf			
14						dref		
15					stbl			
16						stsd		
17							avc1	
18								avcC
19								pasp
20						stts		
21						stss		
22						stsz		
23						stsc		
24						co64		
25		trak						
26			tkhd					
27			mdia					
28				mdhd				
29				hdlr				
30				minf				
31					smhd			
32					dinf			
33						dref		
34					stbl			
35						stsd		
36							mp4a	
37								esds
38						stts		
39						stsz		
40						stsc		
41						co64		
42	free							
43	mdat							

	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			htcb					
6		trak						
7			tkhd					
8			mdia					
9				mdhd				
10				hdlr				
11				minf				
12					vmhd			
13					dinf			
14						dref		
15					stbl			
16						stsd		
17							avc1	
18								avcC
19								pasp
20						stts		
21						stss		
22						stsz		
23						stsc		
24						co64		
25		trak						
26			tkhd					
27			mdia					
28				mdhd				
29				hdlr				
30				minf				
31					smhd			
32					dinf			
33						dref		
34					stbl			
35						stsd		
36							mp4a	
37								esds
38						stts		
39						stsz		
40						stsc		
41						co64		
42	free							
43	mdat							

Figure 33. Comparison of HTC One M7 and HTC One M8 Structure

While the file structures are identical when analyzed with MediaInfo, their metadata begins to reveal differences. Both files are recorded in identical resolution but the File Type Box (‘ftyp’) reveals that the M7 identifies its file with a file type of ‘mp42’ representing the ISO/IEC 14496-14 standard while the M8 identifies with the file type ‘isom’ representing an ISO Base Media file. This should be an immediate cause for the two files to be viewed as originating from different devices but HTC uses a different video format profile in the two devices.

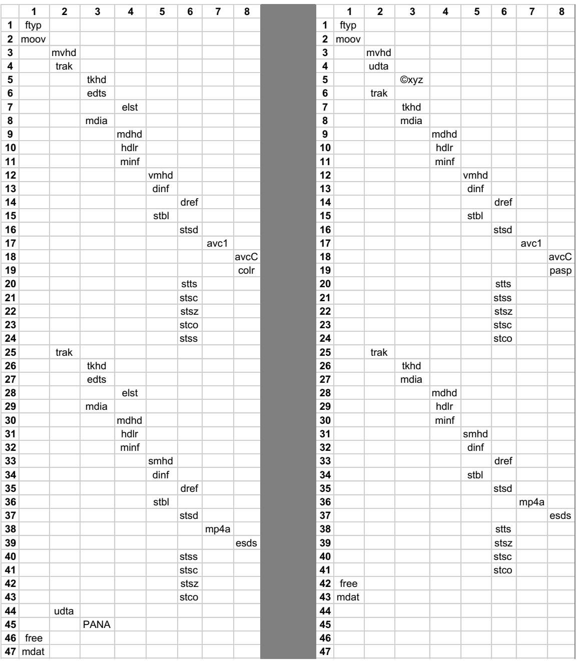
The HTC One M7 reports a video format profile of ‘Baseline @L4’ and the HTC One M8 reports a video format profile of ‘High@L4’.



General		**General**	
Complete name	1920x1080-HTC-One-M7-HD-MC-1.mp4	Complete name	1920x1080-htc_one_m8_01.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	Base Media / Version 2	Format profile	Base Media
Codec ID	mp42	Codec ID	isom
File size	14.1 MiB	File size	48.7 MiB
Duration	5s 504ms	Duration	20s 203ms
Overall bit rate	21.5 Mbps	Overall bit rate	20.2 Mbps
Encoded date	UTC 2015-04-28 00:54:03	Encoded date	UTC 2014-04-03 08:02:33
Tagged date	UTC 2015-04-28 00:54:03	Tagged date	UTC 2014-04-03 08:02:33
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	Baseline@L4	Format profile	High@L4
Format settings, CABAC	No	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=31	Format settings, GOP	M=1, N=60
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	5s 500ms	Duration	20s 195ms
Source duration	5s 506ms	Bit rate	19.7 Mbps
Bit rate	20.1 Mbps	Width	1 920 pixels
Width	1 920 pixels	Height	1 080 pixels
Height	1 080 pixels	Display aspect ratio	16:09
Display aspect ratio	16:09	Frame rate mode	Variable
Rotation	90°	Frame rate	30.354 fps
Frame rate mode	Variable	Minimum frame rate	30.313 fps
Frame rate	29.970 fps	Maximum frame rate	30.395 fps
Minimum frame rate	25.561 fps	Color space	YUV
Maximum frame rate	30.303 fps	Chroma subsampling	4:02:00
Color space	YUV	Bit depth	8 bits
Chroma subsampling	4:02:00	Scan type	Progressive
Bit depth	8 bits	Bits/(Pixel*Frame)	0.313
Scan type	Progressive	Stream size	47.5 MiB (97%)
Bits/(Pixel*Frame)	0.323	Title	VideoHandle
Stream size	13.2 MiB (94%)	Language	English
Source stream size	13.2 MiB (94%)	Encoded date	UTC 2014-04-03 08:02:33
Title	VideoHandle	Tagged date	UTC 2014-04-03 08:02:33
Language	English		
Encoded date	UTC 2015-04-28 00:54:03		
Tagged date	UTC 2015-04-28 00:54:03		
mdhd_Duration	5500		
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	5s 504ms	Duration	20s 203ms
Bit rate mode	Constant	Source duration	20s 209ms
Bit rate	192 Kbps	Bit rate mode	Constant
Nominal bit rate	96.0 Kbps	Bit rate	192 Kbps
Channel(s)	2 channels	Nominal bit rate	96.0 Kbps
Channel positions	Front: L R	Channel(s)	2 channels
Sampling rate	48.0 KHz	Channel positions	Front: L R
Compression mode	Lossy	Sampling rate	48.0 KHz
Stream size	129 KiB (1%)	Compression mode	Lossy
Title	SoundHandle	Stream size	470 KiB (1%)
Language	English	Source stream size	470 KiB (1%)
Encoded date	UTC 2015-04-28 00:54:03	Title	SoundHandle
Tagged date	UTC 2015-04-28 00:54:03	Language	English
		Encoded date	UTC 2014-04-03 08:02:33
		Tagged date	UTC 2014-04-03 08:02:33
		mdhd_Duration	20203

Figure 34. MediaInfo Comparison of HTC One M7 and HTC One M8

Not all devices of identical manufacturer create files of identical structure requiring further analysis. In the case of the two Panasonic Lumix devices analyzed, the structure is enough to differentiate between the two files.

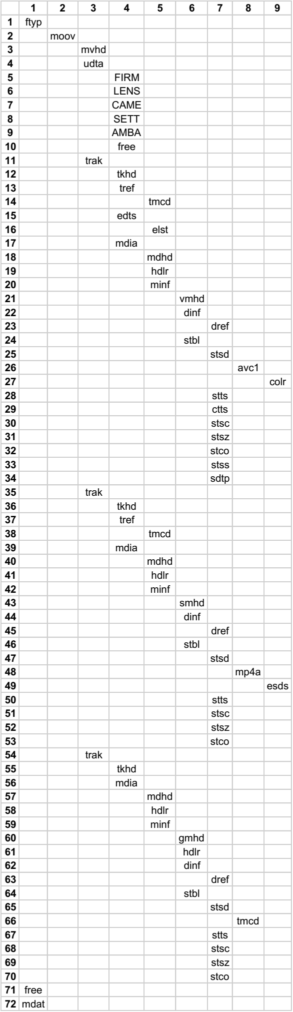


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		trak						
5			tkhd					
6			edts					
7				elst				
8			mdia					
9				mdhd				
10				hdlr				
11				minf				
12					vmhd			
13					dinf			
14						dref		
15					stbl			
16						stsd		
17							avc1	
18								avcC
19								colr
20						stts		
21						stsc		
22						stsz		
23						stco		
24						stss		
25		trak						
26			tkhd					
27			edts					
28				elst				
29			mdia					
30				mdhd				
31				hdlr				
32				minf				
33					smhd			
34					dinf			
35						dref		
36					stbl			
37						stsd		
38							mp4a	
39								esds
40						stss		
41						stsc		
42						stsz		
43						stco		
44		udta						
45			PANA					
46	free							
47	mdat							

	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			©xyz					
6		trak						
7			tkhd					
8			mdia					
9				mdhd				
10				hdlr				
11				minf				
12					vmhd			
13					dinf			
14						dref		
15					stbl			
16						stsd		
17							avc1	
18								avcC
19								pasp
20						stts		
21						stss		
22						stsz		
23						stsc		
24						stco		
25		trak						
26			tkhd					
27			mdia					
28				mdhd				
29				hdlr				
30				minf				
31					smhd			
32					dinf			
33						dref		
34					stbl			
35						stsd		
36							mp4a	
37								esds
38						stts		
39						stsz		
40						stsc		
41						stco		
42	free							
43	mdat							
44								
45								
46								
47								

Figure 35. Comparison of Panasonic Lumix DMC-TS5
and Panasonic Lumix DMC-CM1 Structure

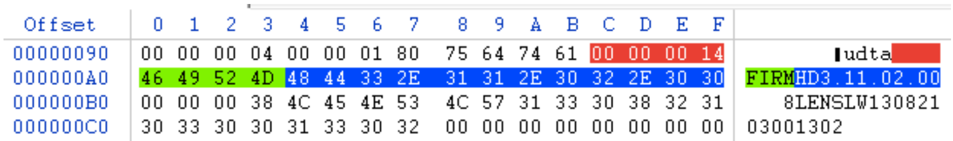
Different devices record different amounts of metadata about the device itself. The devices analyzed so far contain no meaningful amount of metadata about the recording device itself and at best can only be identified by their file structure and metadata. In the case of the GoPro Hero 3, there is a staggering amount of forensically relevant metadata contained within the file structure of every video created on a given device.



	1	2	3	4	5	6	7	8	9
1	ftyp								
2		moov							
3			mvhd						
4			udta						
5				FIRM					
6				LENS					
7				CAME					
8				SETT					
9				AMBA					
10				free					
11			trak						
12				tkhd					
13				tref					
14					tmcd				
15				edts					
16					elst				
17				mdia					
18					mdhd				
19					hdlr				
20					minf				
21						vmhd			
22						dinf			
23							dref		
24						stbl			
25							stsd		
26								avc1	
27									colr
28							stts		
29							ctts		
30							stsc		
31							stsz		
32							stco		
33							stss		
34							sdtp		
35			trak						
36				tkhd					
37				tref					
38					tmcd				
39				mdia					
40					mdhd				
41					hdlr				
42					minf				
43						smhd			
44						dinf			
45							dref		
46						stbl			
47							stsd		
48								mp4a	
49									esds
50							stts		
51							stsc		
52							stsz		
53							stco		
54			trak						
55				tkhd					
56				mdia					
57					mdhd				
58					hdlr				
59					minf				
60						gmhd			
61						hdlr			
62						dinf			
63							dref		
64						stbl			
65							stsd		
66								tmcd	
67							stts		
68							stsc		
69							stsz		
70							stco		
71	free								
72	mdat								

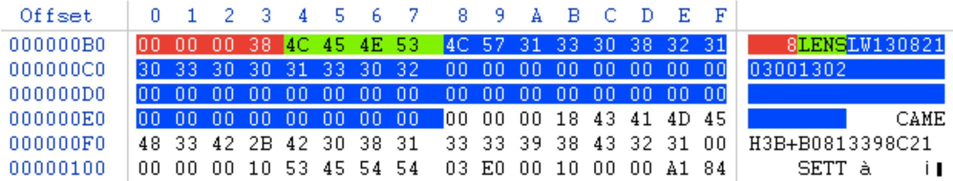
Figure 36. GoPro Hero 3 Structure

Examining the structure of a sample Go Pro Hero 3 file reveals an extensive structure of MPEG-4 Boxes including three instances of a Track Box ('trak') instead of the two that have been observed in other files. The GoPro also includes a number of manufacturer-specific boxes contained in the User Data Box ('udta'). Of increasing interest are the containers 'FIRM', 'LENS', and 'CAME'. While 'FIRM' and 'LENS' both contain useful metadata, 'CAME' simply records the serial number of the device. This is an extraordinary piece of data unique to the GoPro devices examined for this paper.



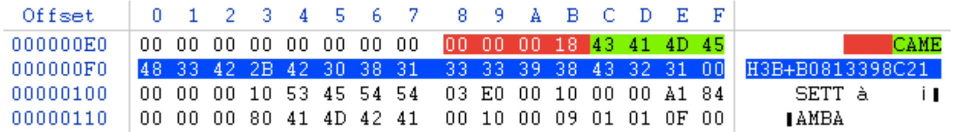
```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000090  00 00 00 04 00 00 01 80  75 64 74 61 00 00 00 14      ▮udta
000000A0  46 49 52 4D 48 44 33 2E  31 31 2E 30 32 2E 30 30  FIRMHD3.11.02.00
000000B0  00 00 00 38 4C 45 4E 53  4C 57 31 33 30 38 32 31     8LENSLW130821
000000C0  30 33 30 30 31 33 30 32  00 00 00 00 00 00 00 00  03001302
```

Figure 37. Parsing GoPro FIRM Box



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
000000B0  00 00 00 38 4C 45 4E 53  4C 57 31 33 30 38 32 31     8LENSLW130821
000000C0  30 33 30 30 31 33 30 32  00 00 00 00 00 00 00 00  03001302
000000D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00  00 00 00 18 43 41 4D 45              CAME
000000F0  48 33 42 2B 42 30 38 31  33 33 39 38 43 32 31 00  H3B+B0813398C21
00000100  00 00 00 10 53 45 54 54  03 E0 00 10 00 00 A1 84      SETT à     i▮
```

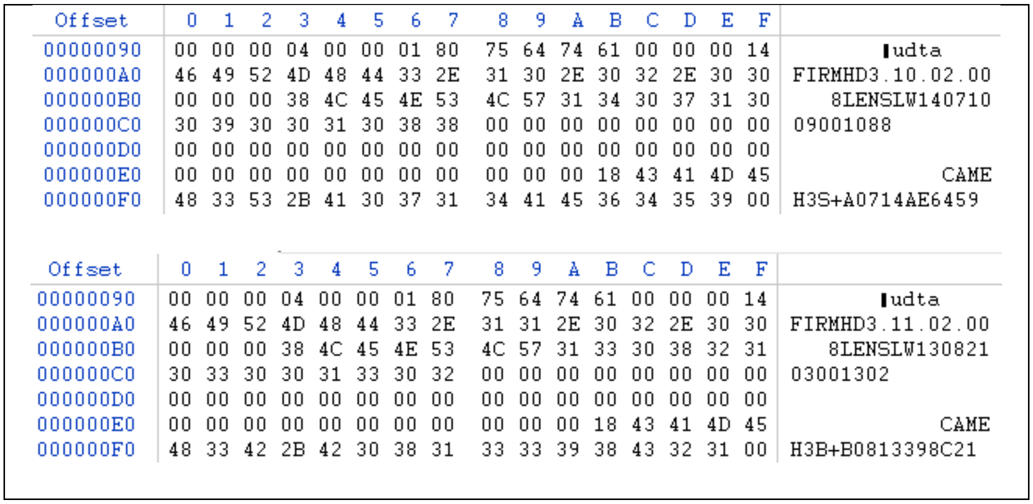
Figure 38. Parsing GoPro LENS Box



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
000000E0  00 00 00 00 00 00 00 00  00 00 00 18 43 41 4D 45              CAME
000000F0  48 33 42 2B 42 30 38 31  33 33 39 38 43 32 31 00  H3B+B0813398C21
00000100  00 00 00 10 53 45 54 54  03 E0 00 10 00 00 A1 84      SETT à     i▮
00000110  00 00 00 80 41 4D 42 41  00 10 00 09 01 01 0F 00     ▮AMBA
```

Figure 39. Parsing GoPro CAME Box

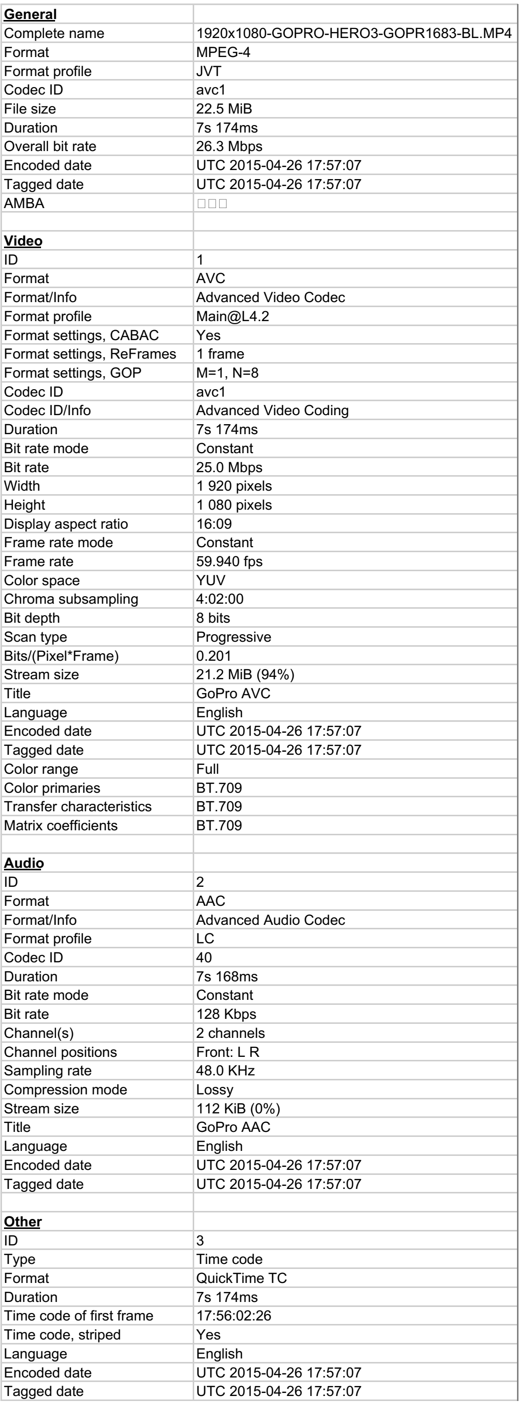
In order to demonstrate the unique nature of the ‘CAME’ box, the User Data Box (‘udta’) of two different model Go Pro devices were compared to show the unique nature of the ‘CAME’ box and its ability to identify the model and serial number of each device.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000090  00 00 00 04 00 00 01 80  75 64 74 61 00 00 00 14      ▮udta
000000A0  46 49 52 4D 48 44 33 2E  31 30 2E 30 32 2E 30 30  FIRMHD3.10.02.00
000000B0  00 00 00 38 4C 45 4E 53  4C 57 31 34 30 37 31 30     8LENSLW140710
000000C0  30 39 30 30 31 30 38 38  00 00 00 00 00 00 00 00  09001088
000000D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00  00 00 00 18 43 41 4D 45              CAME
000000F0  48 33 53 2B 41 30 37 31  34 41 45 36 34 35 39 00  H3S+A0714AE6459

Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000090  00 00 00 04 00 00 01 80  75 64 74 61 00 00 00 14      ▮udta
000000A0  46 49 52 4D 48 44 33 2E  31 31 2E 30 32 2E 30 30  FIRMHD3.11.02.00
000000B0  00 00 00 38 4C 45 4E 53  4C 57 31 33 30 38 32 31     8LENSLW130821
000000C0  30 33 30 30 31 33 30 32  00 00 00 00 00 00 00 00  03001302
000000D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00  00 00 00 18 43 41 4D 45              CAME
000000F0  48 33 42 2B 42 30 38 31  33 33 39 38 43 32 31 00  H3B+B0813398C21
```

Figure 40. Comparison of two different GoPro User Data Boxes (‘udta’)

Analyzing the example GoPro file with MediaInfo reveals a number of self-identifying properties referring to the GoPro by name as well as more information about the third Track Box (‘trak’). This box contains a QuickTime time code track which is unique to the GoPro among the devices examined for this paper.

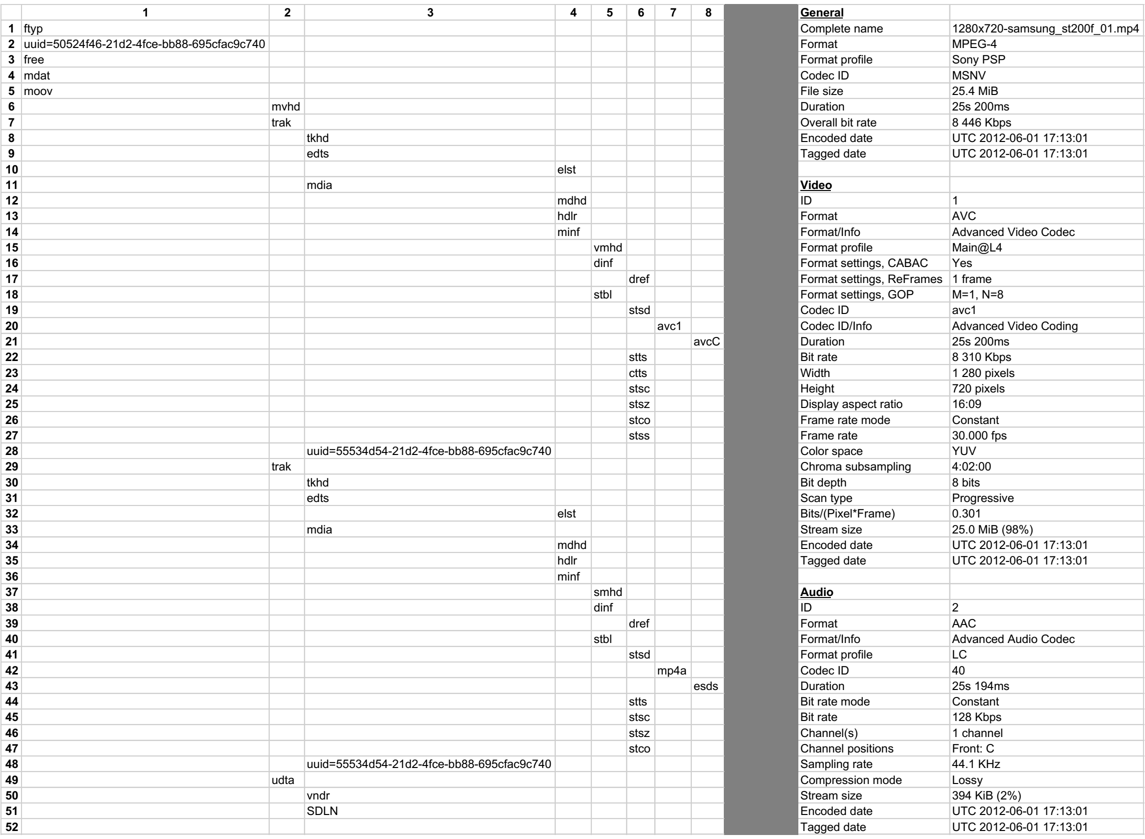


General	
Complete name	1920x1080-GOPRO-HERO3-GOPR1683-BL.MP4
Format	MPEG-4
Format profile	JVT
Codec ID	avc1
File size	22.5 MiB
Duration	7s 174ms
Overall bit rate	26.3 Mbps
Encoded date	UTC 2015-04-26 17:57:07
Tagged date	UTC 2015-04-26 17:57:07
AMBA	□□□
Video	
ID	1
Format	AVC
Format/Info	Advanced Video Codec
Format profile	Main@L4.2
Format settings, CABAC	Yes
Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=8
Codec ID	avc1
Codec ID/Info	Advanced Video Coding
Duration	7s 174ms
Bit rate mode	Constant
Bit rate	25.0 Mbps
Width	1 920 pixels
Height	1 080 pixels
Display aspect ratio	16:09
Frame rate mode	Constant
Frame rate	59.940 fps
Color space	YUV
Chroma subsampling	4:02:00
Bit depth	8 bits
Scan type	Progressive
Bits/(Pixel*Frame)	0.201
Stream size	21.2 MiB (94%)
Title	GoPro AVC
Language	English
Encoded date	UTC 2015-04-26 17:57:07
Tagged date	UTC 2015-04-26 17:57:07
Color range	Full
Color primaries	BT.709
Transfer characteristics	BT.709
Matrix coefficients	BT.709
Audio	
ID	2
Format	AAC
Format/Info	Advanced Audio Codec
Format profile	LC
Codec ID	40
Duration	7s 168ms
Bit rate mode	Constant
Bit rate	128 Kbps
Channel(s)	2 channels
Channel positions	Front: L R
Sampling rate	48.0 KHz
Compression mode	Lossy
Stream size	112 KiB (0%)
Title	GoPro AAC
Language	English
Encoded date	UTC 2015-04-26 17:57:07
Tagged date	UTC 2015-04-26 17:57:07
Other	
ID	3
Type	Time code
Format	QuickTime TC
Duration	7s 174ms
Time code of first frame	17:56:02:26
Time code, striped	Yes
Language	English
Encoded date	UTC 2015-04-26 17:57:07
Tagged date	UTC 2015-04-26 17:57:07

Figure 41. GoPro Hero 3 MediaInfo Analysis

In addition to the identifying serial numbers contained in the metadata of the GoPro recordings, if an owner has entered their name in the camera menu this information will also be displayed in the User Data Box (‘udta’). In the research for this paper there were no tools that will parse out the User Data Box (‘udta’) box of a GoPro recording. This remarkably valuable information can only be found by parsing the file manually using a hex editor.

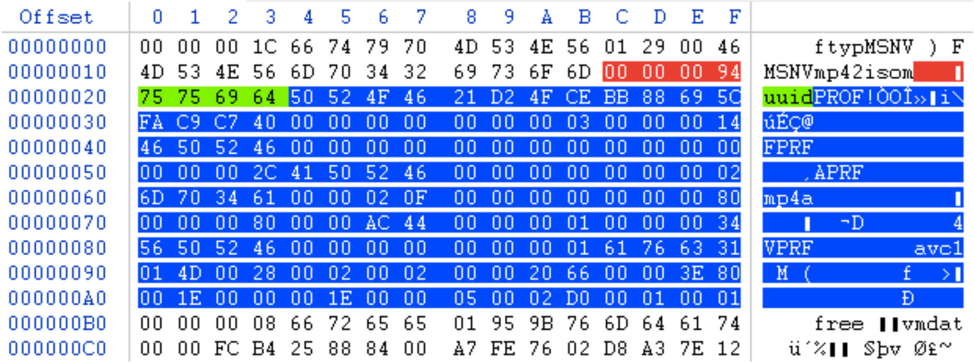
When using AtomicParsley to analyze the Samsung ST200F, a number of UUID’s are returned as part of the file structure: 50524f46-21d2-4fce-bb88-695cfac9c740 contained in the top level of the file, and two instances of 55534d54-21d2-4fce-bb88-695cfac9c740 occurring once in each of the two Trak Boxes (‘trak’). Atomic Parsley returns the UUID as a box identified with the prefix “uuid=” and returns the formatted UUID as part of its standard output. In order to analyze the UUID’s present in the video from the Samsung ST200F, the output of MediaInfo was examined to specifically establish a baseline of the encoding date and time. Since a UUID could possibly represent time and a MAC address[19], it would be an important development if the embedded data contained meaningful data regarding the time and date of the recording and possibly a unique identifying number of the recording device itself.



	1	2	3	4	5	6	7	8		**General**	
1	ftyp									Complete name	1280x720-samsung_st200f_01.mp4
2	uuid=50524f46-21d2-4fce-bb88-695cfac9c740									Format	MPEG-4
3	free									Format profile	Sony PSP
4	mdat									Codec ID	MSNV
5	moov									File size	25.4 MiB
6		mvhd								Duration	25s 200ms
7		trak								Overall bit rate	8 446 Kbps
8			tkhd							Encoded date	UTC 2012-06-01 17:13:01
9			edts							Tagged date	UTC 2012-06-01 17:13:01
10				elst							
11			mdia							**Video**	
12				mdhd						ID	1
13				hdlr						Format	AVC
14				minf						Format/Info	Advanced Video Codec
15					vmhd					Format profile	Main@L4
16					dinf					Format settings, CABAC	Yes
17						dref				Format settings, ReFrames	1 frame
18					stbl					Format settings, GOP	M=1, N=8
19						stsd				Codec ID	avc1
20							avc1			Codec ID/Info	Advanced Video Coding
21								avcC		Duration	25s 200ms
22						stts				Bit rate	8 310 Kbps
23						ctts				Width	1 280 pixels
24						stsc				Height	720 pixels
25						stsz				Display aspect ratio	16:09
26						stco				Frame rate mode	Constant
27						stss				Frame rate	30.000 fps
28			uuid=55534d54-21d2-4fce-bb88-695cfac9c740							Color space	YUV
29		trak								Chroma subsampling	4:02:00
30			tkhd							Bit depth	8 bits
31			edts							Scan type	Progressive
32				elst						Bits/(Pixel*Frame)	0.301
33			mdia							Stream size	25.0 MiB (98%)
34				mdhd						Encoded date	UTC 2012-06-01 17:13:01
35				hdlr						Tagged date	UTC 2012-06-01 17:13:01
36				minf							
37					smhd					**Audio**	
38					dinf					ID	2
39						dref				Format	AAC
40					stbl					Format/Info	Advanced Audio Codec
41						stsd				Format profile	LC
42							mp4a			Codec ID	40
43								esds		Duration	25s 194ms
44						stts				Bit rate mode	Constant
45						stsc				Bit rate	128 Kbps
46						stsz				Channel(s)	1 channel
47						stco				Channel positions	Front: C
48			uuid=55534d54-21d2-4fce-bb88-695cfac9c740							Sampling rate	44.1 KHz
49		udta								Compression mode	Lossy
50			vndr							Stream size	394 KiB (2%)
51			SDLN							Encoded date	UTC 2012-06-01 17:13:01
52										Tagged date	UTC 2012-06-01 17:13:01

Figure 42. Samsung ST200F Structure and MediaInfo Analysis

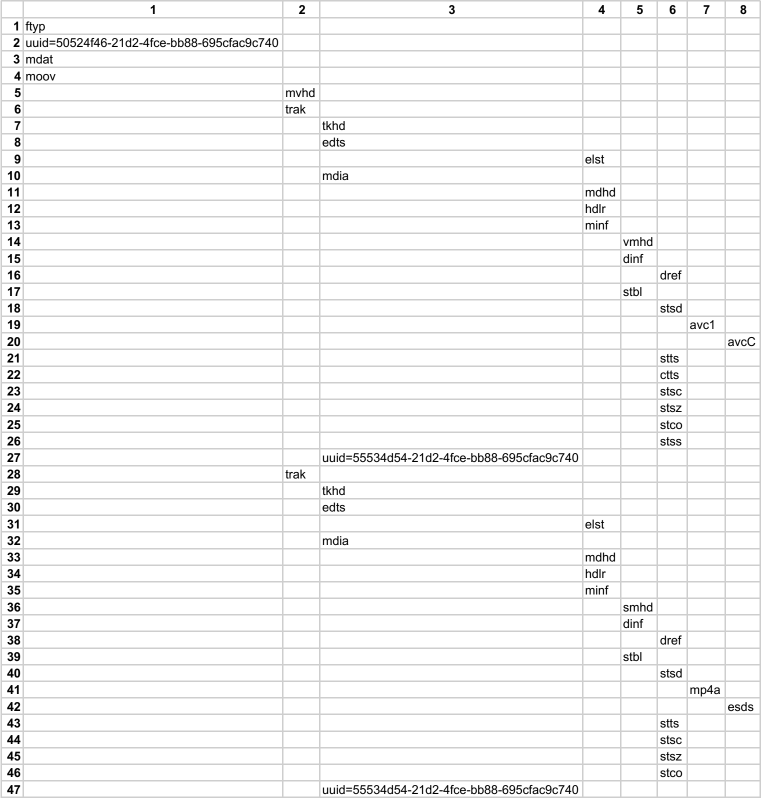
No meaningful connection was discovered between the UUID data returned by AtomicParsley and the embedded timestamps contained within the MPEG-4 structure of the file, it is worth examining the UUID box that AtomicParsley is identifying in this sample file. The AtomicParsley output can be verified with a hexadecimal analysis of the file. In this case, the box structure of the UUID box is correctly formatted with 0x04 bytes representing the box size of 0x94 bytes, a box name of ‘uuid’, followed by the content of the box. In this example, the hexadecimal 0x50524F4621D24FCEBB88695CFAC9C740 is the string being interpreted as the UUID by AtomicParsley. Other meaningful pieces of this box include ‘mp4a’ at offset 0x60 and ‘avc1’ at offset 0x8C but neither offer any insight into the meaning of the UUID included in this file.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 1C 66 74 79 70  4D 53 4E 56 01 29 00 46      ftypMSNV ) F
00000010  4D 53 4E 56 6D 70 34 32  69 73 6F 6D 00 00 00 94  MSNVmp42isom   ▌
00000020  75 75 69 64 50 52 4F 46  21 D2 4F CE BB 88 69 5C  uuidPROF!ÒOÎ»▌i\
00000030  FA C9 C7 40 00 00 00 00  00 00 00 03 00 00 00 14  úÉÇ@
00000040  46 50 52 46 00 00 00 00  00 00 00 00 00 00 00 00  FPRF
00000050  00 00 00 2C 41 50 52 46  00 00 00 00 00 00 00 02     ,APRF
00000060  6D 70 34 61 00 00 02 0F  00 00 00 00 00 00 00 80  mp4a           ▌
00000070  00 00 00 80 00 00 AC 44  00 00 00 01 00 00 00 34     ▌  ¬D       4
00000080  56 50 52 46 00 00 00 00  00 00 00 01 61 76 63 31  VPRF        avc1
00000090  01 4D 00 28 00 02 00 02  00 00 20 66 00 00 3E 80   M (        f  >▌
000000A0  00 1E 00 00 00 1E 00 00  05 00 02 D0 00 01 00 01            Ð
000000B0  00 00 00 08 66 72 65 65  01 95 9B 76 6D 64 61 74      free ▌▌vmdat
000000C0  00 00 FC B4 25 88 84 00  A7 FE 76 02 D8 A3 7E 12    ü´%▌▌ §þv Ø£~
```

Figure 43. Samsung ST200F UUID Hexadecimal Analysis

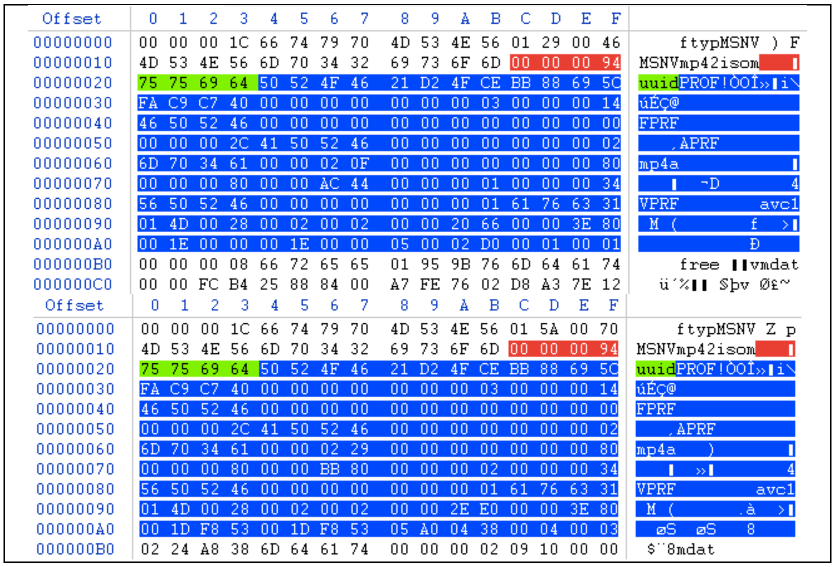
The Sony Cybershot DSC-QX10, another camera examined for this paper, included a series of UUID's. The DSC-QX10 contained three UUID's as part of its file structure, just as the Samsung ST200F did, but the UUID's aren't just in the same positions in the structure of the file the UUID's are identical to those contained in the Samsung ST200F file.



	1	2	3	4	5	6	7	8
1	ftyp							
2	uuid=50524f46-21d2-4fce-bb88-695cfac9c740							
3	mdat							
4	moov							
5		mvhd						
6		trak						
7			tkhd					
8			edts					
9				elst				
10			mdia					
11				mdhd				
12				hdlr				
13				minf				
14					vmhd			
15					dinf			
16						dref		
17					stbl			
18						stsd		
19							avc1	
20								avcC
21						stts		
22						ctts		
23						stsc		
24						stsz		
25						stco		
26						stss		
27			uuid=55534d54-21d2-4fce-bb88-695cfac9c740					
28		trak						
29			tkhd					
30			edts					
31				elst				
32			mdia					
33				mdhd				
34				hdlr				
35				minf				
36					smhd			
37					dinf			
38						dref		
39					stbl			
40						stsd		
41							mp4a	
42								esds
43						stts		
44						stsc		
45						stsz		
46						stco		
47			uuid=55534d54-21d2-4fce-bb88-695cfac9c740					

Figure 44. Sony Cybershot DSC-QX10 Structure

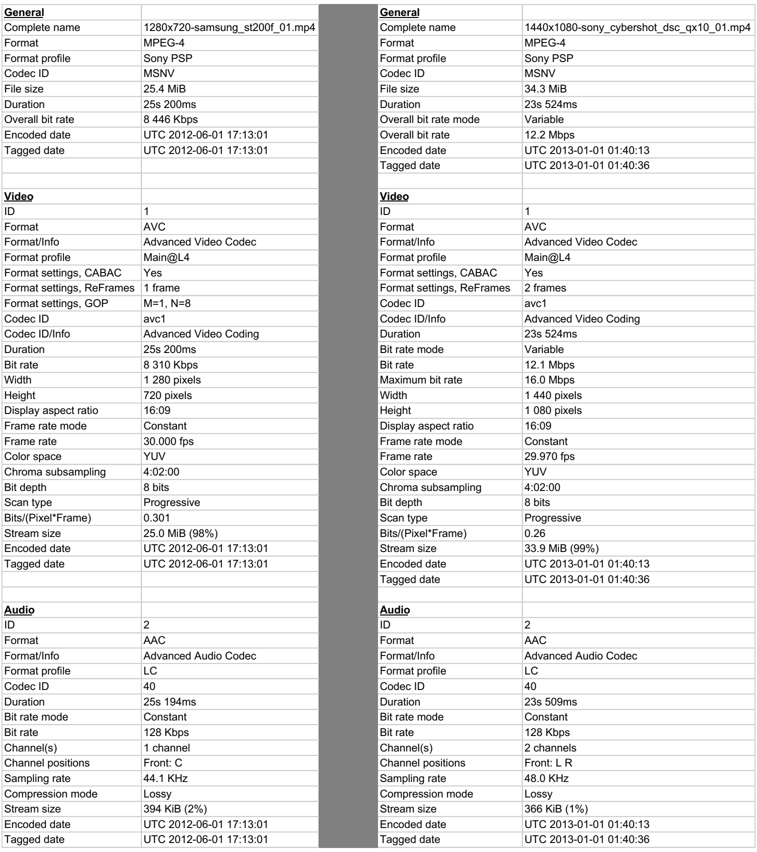
A comparison of the two sample files from the Samsung ST200F and Sony Cybershot DSC-QX10 shows that the hexadecimal structure of what is being interpreted as the UUID at the top level of the file, along with the rest of the contents of that box, is identical.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 1C 66 74 79 70  4D 53 4E 56 01 29 00 46      ftypMSNV ) F
00000010  4D 53 4E 56 6D 70 34 32  69 73 6F 6D 00 00 00 94  MSNVmp42isom   ▌
00000020  75 75 69 64 50 52 4F 46  21 D2 4F CE BB 88 69 5C  uuidPROF!ÒOÎ»▌i\
00000030  FA C9 C7 40 00 00 00 00  00 00 00 03 00 00 00 14  úÉÇ@
00000040  46 50 52 46 00 00 00 00  00 00 00 00 00 00 00 00  FPRF
00000050  00 00 00 2C 41 50 52 46  00 00 00 00 00 00 00 02     ,APRF
00000060  6D 70 34 61 00 00 02 0F  00 00 00 00 00 00 00 80  mp4a           ▌
00000070  00 00 00 80 00 00 AC 44  00 00 00 01 00 00 00 34     ▌  ¬D       4
00000080  56 50 52 46 00 00 00 00  00 00 00 01 61 76 63 31  VPRF        avc1
00000090  01 4D 00 28 00 02 00 02  00 00 20 66 00 00 3E 80   M (        f  >▌
000000A0  00 1E 00 00 00 1E 00 00  05 00 02 D0 00 01 00 01             Ð
000000B0  00 00 00 08 66 72 65 65  01 95 9B 76 6D 64 61 74      free ▌▌vmdat
000000C0  00 00 FC B4 25 88 84 00  A7 FE 76 02 D8 A3 7E 12    ü´%▌▌ §þv Ø£~
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 1C 66 74 79 70  4D 53 4E 56 01 5A 00 70      ftypMSNV Z p
00000010  4D 53 4E 56 6D 70 34 32  69 73 6F 6D 00 00 00 94  MSNVmp42isom   ▌
00000020  75 75 69 64 50 52 4F 46  21 D2 4F CE BB 88 69 5C  uuidPROF!ÒOÎ»▌i\
00000030  FA C9 C7 40 00 00 00 00  00 00 00 03 00 00 00 14  úÉÇ@
00000040  46 50 52 46 00 00 00 00  00 00 00 00 00 00 00 00  FPRF
00000050  00 00 00 2C 41 50 52 46  00 00 00 00 00 00 00 02     ,APRF
00000060  6D 70 34 61 00 00 02 29  00 00 00 00 00 00 00 80  mp4a   )       ▌
00000070  00 00 00 80 00 00 BB 80  00 00 00 02 00 00 00 34     ▌  »▌       4
00000080  56 50 52 46 00 00 00 00  00 00 00 01 61 76 63 31  VPRF        avc1
00000090  01 4D 00 28 00 02 00 02  00 00 2E E0 00 00 3E 80   M (       .à  >▌
000000A0  00 1D F8 53 00 1D F8 53  05 A0 04 38 00 04 00 03    øS  øS    8
000000B0  02 24 A8 38 6D 64 61 74  00 00 00 02 09 10 00 00   $¨8mdat
```

Figure 45. Comparison of Samsung ST200F and Sony Cybershot DSC-QX10 UUID

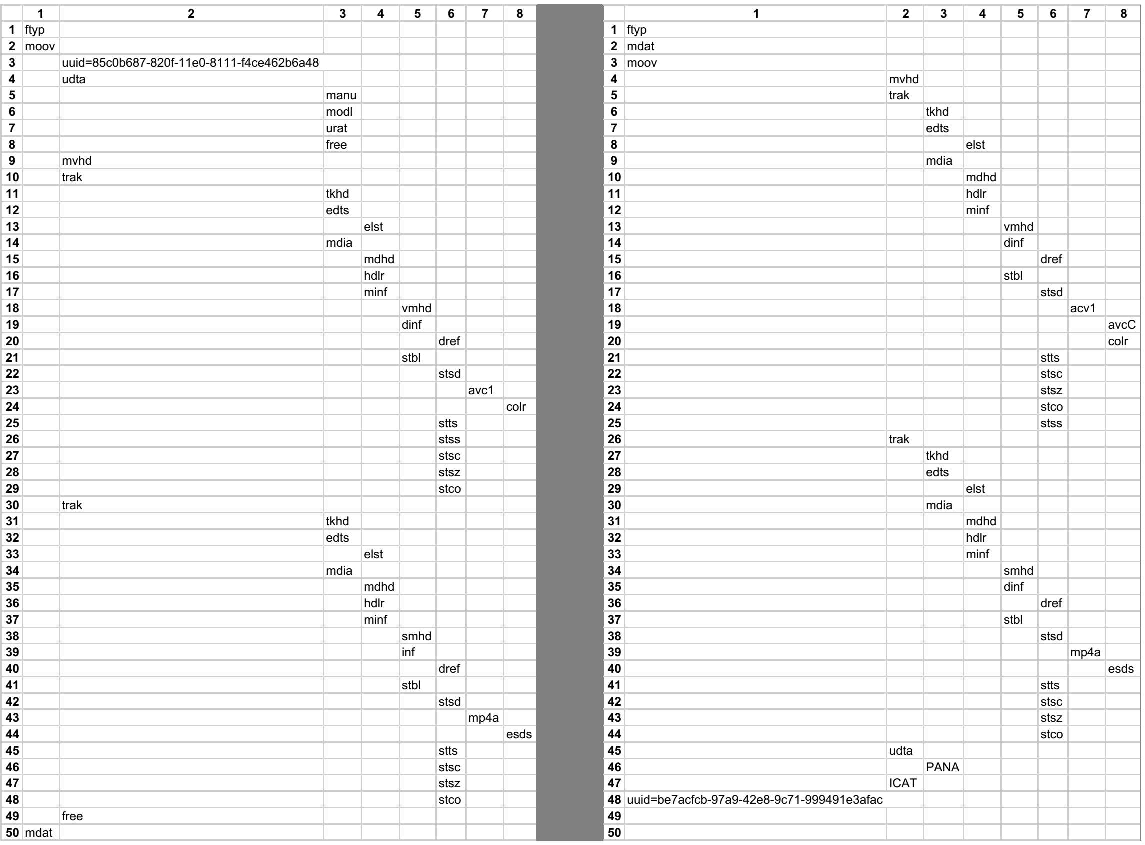
A comparison of the two sample files in MediaInfo reveals that both files that share a common series of UUID’s also share a Codec ID of MSNV. This codec is defined by the MPEG-4 Registration Authority as being for the Sony PlayStation Portable. Further analysis is necessary to confirm the theory that these UUID’s are placed in the file structure in order to support the Sony PlayStation Portable but, in the files collected for this paper, these were the only two devices that created files in this format. It should be noted that regardless of the UUID’s present, these two files can still be differentiated between one another based on their respective file structures and the presence or absence of the ‘free’ box which exists in files created by the Samsung ST200F but not in the Sony Cybershot DSC-QX10.



General			**General**	
Complete name	1280x720-samsung_st200f_01.mp4		Complete name	1440x1080-sony_cybershot_dsc_qx10_01.mp4
Format	MPEG-4		Format	MPEG-4
Format profile	Sony PSP		Format profile	Sony PSP
Codec ID	MSNV		Codec ID	MSNV
File size	25.4 MiB		File size	34.3 MiB
Duration	25s 200ms		Duration	23s 524ms
Overall bit rate	8 446 Kbps		Overall bit rate mode	Variable
Encoded date	UTC 2012-06-01 17:13:01		Overall bit rate	12.2 Mbps
Tagged date	UTC 2012-06-01 17:13:01		Encoded date	UTC 2013-01-01 01:40:13
			Tagged date	UTC 2013-01-01 01:40:36
Video			**Video**	
ID	1		ID	1
Format	AVC		Format	AVC
Format/Info	Advanced Video Codec		Format/Info	Advanced Video Codec
Format profile	Main@L4		Format profile	Main@L4
Format settings, CABAC	Yes		Format settings, CABAC	Yes
Format settings, ReFrames	1 frame		Format settings, ReFrames	2 frames
Format settings, GOP	M=1, N=8		Codec ID	avc1
Codec ID	avc1		Codec ID/Info	Advanced Video Coding
Codec ID/Info	Advanced Video Coding		Duration	23s 524ms
Duration	25s 200ms		Bit rate mode	Variable
Bit rate	8 310 Kbps		Bit rate	12.1 Mbps
Width	1 280 pixels		Maximum bit rate	16.0 Mbps
Height	720 pixels		Width	1 440 pixels
Display aspect ratio	16:09		Height	1 080 pixels
Frame rate mode	Constant		Display aspect ratio	16:09
Frame rate	30.000 fps		Frame rate mode	Constant
Color space	YUV		Frame rate	29.970 fps
Chroma subsampling	4:02:00		Color space	YUV
Bit depth	8 bits		Chroma subsampling	4:02:00
Scan type	Progressive		Bit depth	8 bits
Bits/(Pixel*Frame)	0.301		Scan type	Progressive
Stream size	25.0 MiB (98%)		Bits/(Pixel*Frame)	0.26
Encoded date	UTC 2012-06-01 17:13:01		Stream size	33.9 MiB (99%)
Tagged date	UTC 2012-06-01 17:13:01		Encoded date	UTC 2013-01-01 01:40:13
			Tagged date	UTC 2013-01-01 01:40:36
Audio			**Audio**	
ID	2		ID	2
Format	AAC		Format	AAC
Format/Info	Advanced Audio Codec		Format/Info	Advanced Audio Codec
Format profile	LC		Format profile	LC
Codec ID	40		Codec ID	40
Duration	25s 194ms		Duration	23s 509ms
Bit rate mode	Constant		Bit rate mode	Constant
Bit rate	128 Kbps		Bit rate	128 Kbps
Channel(s)	1 channel		Channel(s)	2 channels
Channel positions	Front: C		Channel positions	Front: L R
Sampling rate	44.1 KHz		Sampling rate	48.0 KHz
Compression mode	Lossy		Compression mode	Lossy
Stream size	394 KiB (2%)		Stream size	366 KiB (1%)
Encoded date	UTC 2012-06-01 17:13:01		Encoded date	UTC 2013-01-01 01:40:13
Tagged date	UTC 2012-06-01 17:13:01		Tagged date	UTC 2013-01-01 01:40:36

Figure 46. MediaInfo Comparison of Samsung ST200F and Sony Cybershot DSC-QX10

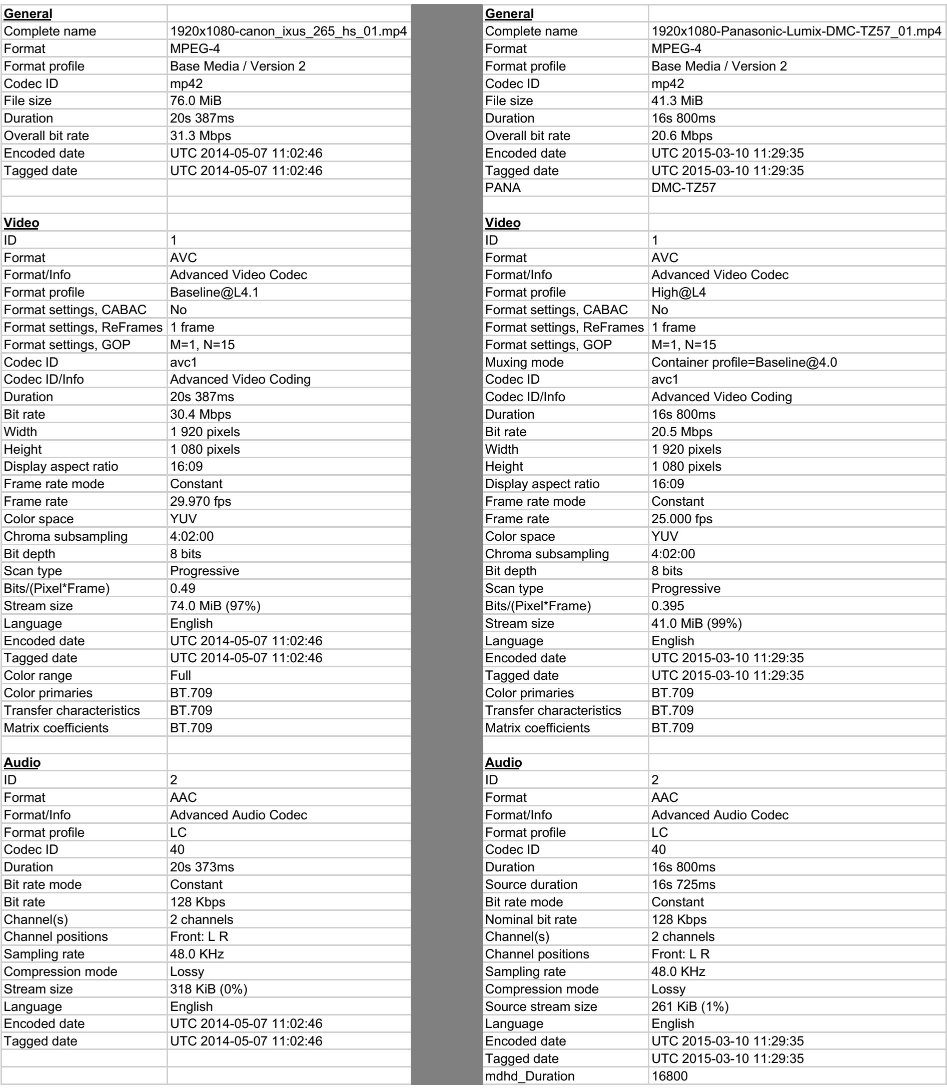
The Samsung ST200F and Sony Cybershot are not the only devices with UUID’s examined for this paper. Two other devices contained UUID’s: Canon IXUS 265 and the Panasonic Lumix DMC-TZ57. A comparison of their file structures reveals that they are distinguishable from one another based on their MPEG-4 box structures and they contain UUID’s which are unique to each respective device.



	1	2	3	4	5	6	7	8			1	2	3	4	5	6	7	8
1	ftyp									1	ftyp							
2	moov									2	mdat							
3		uuid=85c0b687-820f-11e0-8111-f4ce462b6a48								3	moov							
4		udta								4		mvhd						
5			manu							5		trak						
6			modl							6			tkhd					
7			urat							7			edts					
8			free							8				elst				
9		mvhd								9			mdia					
10		trak								10				mdhd				
11			tkhd							11				hdlr				
12			edts							12				minf				
13				elst						13					vmhd			
14			mdia							14					dinf			
15				mdhd						15						dref		
16				hdlr						16					stbl			
17				minf						17						stsd		
18					vmhd					18							acv1	
19					dinf					19								avcC
20						dref				20								colr
21					stbl					21						stts		
22						stsd				22						stsc		
23							avc1			23						stsz		
24								colr		24						stco		
25						stts				25						stss		
26						stss				26		trak						
27						stsc				27			tkhd					
28						stsz				28			edts					
29						stco				29				elst				
30		trak								30			mdia					
31			tkhd							31				mdhd				
32			edts							32				hdlr				
33				elst						33				minf				
34			mdia							34					smhd			
35				mdhd						35					dinf			
36				hdlr						36						dref		
37				minf						37					stbl			
38					smhd					38						stsd		
39					inf					39							mp4a	
40						dref				40								esds
41					stbl					41						stts		
42						stsd				42						stsc		
43							mp4a			43						stsz		
44								esds		44						stco		
45						stts				45		udta						
46						stsc				46			PANA					
47						stsz				47		ICAT						
48						stco				48	uuid=be7acfcb-97a9-42e8-9c71-999491e3afac							
49		free								49								
50	mdat									50								

Figure 47. Comparison of Canon IXUS 265 and Panasonic Lumix DMC-TZ57 Structure

Unfortunately, neither of these UUID's contained a timestamp that matched the embedded timestamps in the MPEG-4 standard. MediaInfo returns data which helps to support the differentiation between the two files but adds no support for the correlation between the properties of the files, as it did with Sony PlayStation Portable formatting in the cases of the Samsung ST200F and the Sony Cybershot DSC-QX10. When comparing these two files it is important to note that while their file structures showed clear differences between the two files their reports from MediaInfo were remarkably similar.



General		General	
Complete name	1920x1080-canon_ixus_265_hs_01.mp4	Complete name	1920x1080-Panasonic-Lumix-DMC-TZ57_01.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	Base Media / Version 2	Format profile	Base Media / Version 2
Codec ID	mp42	Codec ID	mp42
File size	76.0 MiB	File size	41.3 MiB
Duration	20s 387ms	Duration	16s 800ms
Overall bit rate	31.3 Mbps	Overall bit rate	20.6 Mbps
Encoded date	UTC 2014-05-07 11:02:46	Encoded date	UTC 2015-03-10 11:29:35
Tagged date	UTC 2014-05-07 11:02:46	Tagged date	UTC 2015-03-10 11:29:35
		PANA	DMC-TZ57
Video		Video	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	Baseline@L4.1	Format profile	High@L4
Format settings, CABAC	No	Format settings, CABAC	No
Format settings, ReFrames	1 frame	Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=15	Format settings, GOP	M=1, N=15
Codec ID	avc1	Muxing mode	Container profile=Baseline@4.0
Codec ID/Info	Advanced Video Coding	Codec ID	avc1
Duration	20s 387ms	Codec ID/Info	Advanced Video Coding
Bit rate	30.4 Mbps	Duration	16s 800ms
Width	1 920 pixels	Bit rate	20.5 Mbps
Height	1 080 pixels	Width	1 920 pixels
Display aspect ratio	16:09	Height	1 080 pixels
Frame rate mode	Constant	Display aspect ratio	16:09
Frame rate	29.970 fps	Frame rate mode	Constant
Color space	YUV	Frame rate	25.000 fps
Chroma subsampling	4:02:00	Color space	YUV
Bit depth	8 bits	Chroma subsampling	4:02:00
Scan type	Progressive	Bit depth	8 bits
Bits/(Pixel*Frame)	0.49	Scan type	Progressive
Stream size	74.0 MiB (97%)	Bits/(Pixel*Frame)	0.395
Language	English	Stream size	41.0 MiB (99%)
Encoded date	UTC 2014-05-07 11:02:46	Language	English
Tagged date	UTC 2014-05-07 11:02:46	Encoded date	UTC 2015-03-10 11:29:35
Color range	Full	Tagged date	UTC 2015-03-10 11:29:35
Color primaries	BT.709	Color primaries	BT.709
Transfer characteristics	BT.709	Transfer characteristics	BT.709
Matrix coefficients	BT.709	Matrix coefficients	BT.709
Audio		Audio	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	20s 373ms	Duration	16s 800ms
Bit rate mode	Constant	Source duration	16s 725ms
Bit rate	128 Kbps	Bit rate mode	Constant
Channel(s)	2 channels	Nominal bit rate	128 Kbps
Channel positions	Front: L R	Channel(s)	2 channels
Sampling rate	48.0 KHz	Channel positions	Front: L R
Compression mode	Lossy	Sampling rate	48.0 KHz
Stream size	318 KiB (0%)	Compression mode	Lossy
Language	English	Source stream size	261 KiB (1%)
Encoded date	UTC 2014-05-07 11:02:46	Language	English
Tagged date	UTC 2014-05-07 11:02:46	Encoded date	UTC 2015-03-10 11:29:35
		Tagged date	UTC 2015-03-10 11:29:35
		mdhd_Duration	16800

Figure 48. MediaInfo Comparison of Canon IXUS 265 and Panasonic Lumix DMC-TZ57

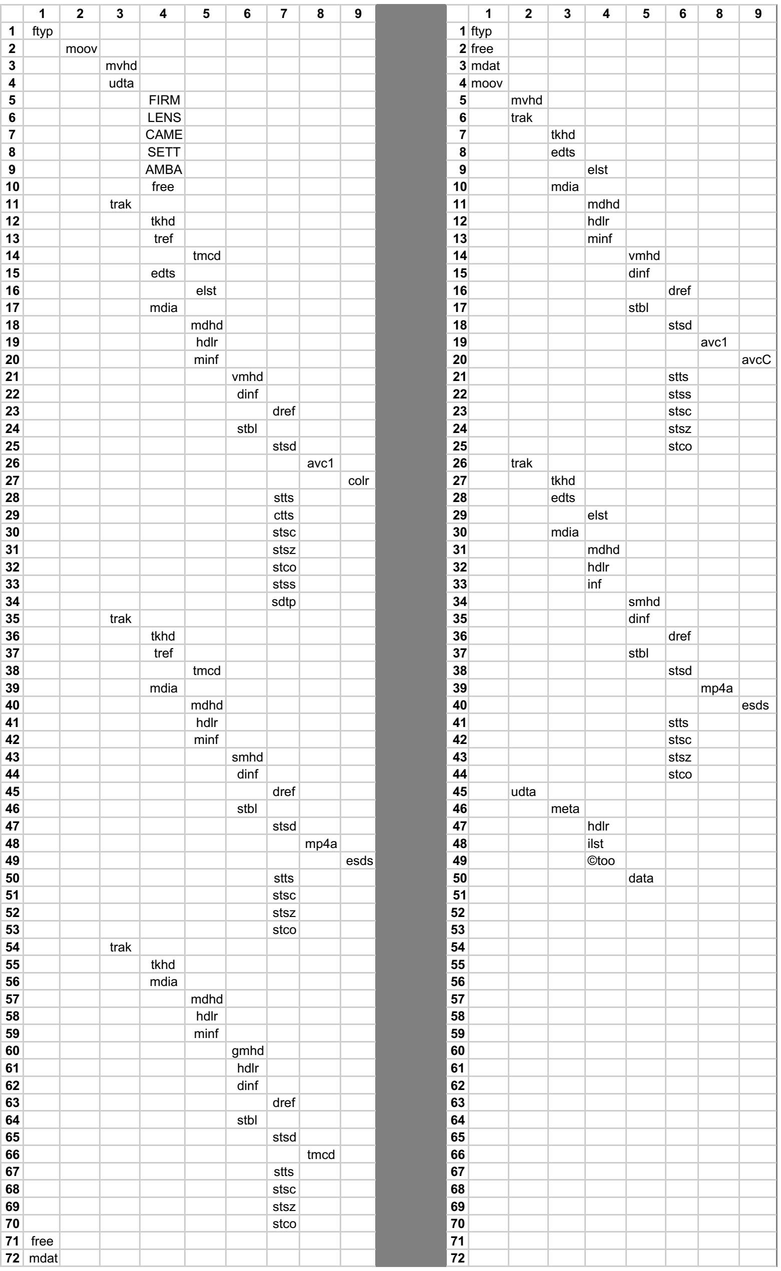
CHAPTER VI

ANALYSIS OF EDITED FILES

The files examined for this paper that contain the most forensically relevant data are by far those created by the GoPro devices. Being able to identify which make and model of camera a file was created on is one thing but having the recorded evidence of a serial number of the device in question is invaluable. Whether the file being examined came from a GoPro device or from another device that records no meaningful user data, the structure of a file is changed when it is re-encoded. For the purposes of this testing, no edits were made to the contents of the video itself. Sample files from a GoPro and the LG G3 were simply re-encoded using commonly available software tools, being careful to match software settings to export in the MPEG-4 format for each video editing tool. These resulting files were then analyzed using AtomicParsley and MediaInfo to demonstrate the results of this re-encoding.

ffmpeg

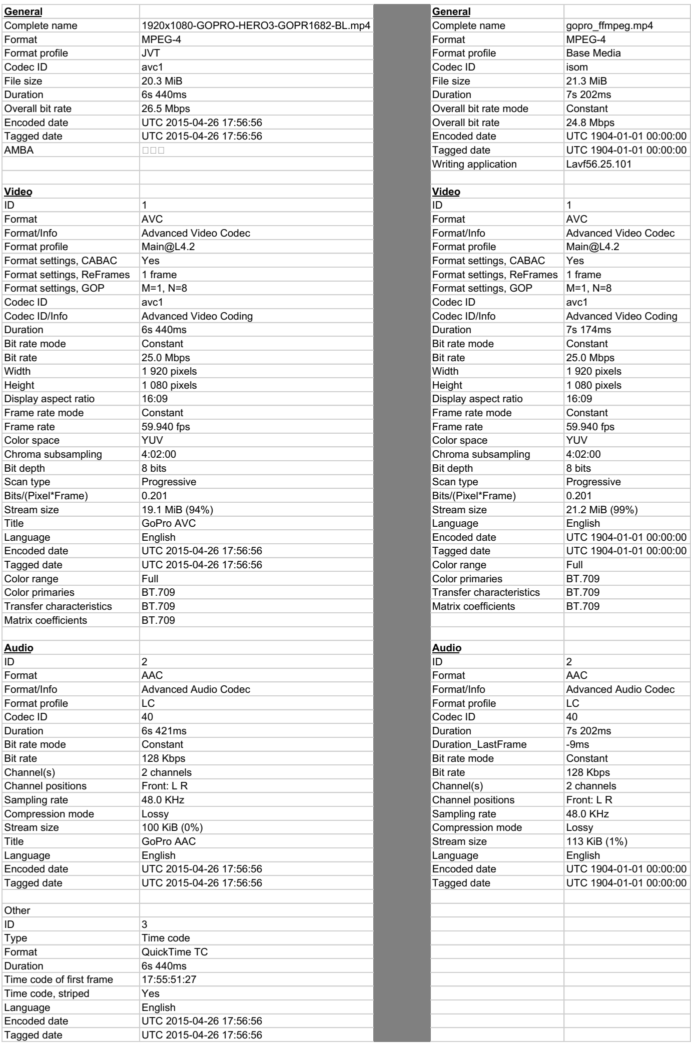
The first tool tested was ffmpeg, a piece of software released under the GNU General Public License. It is a powerful audio and video encoder and decoder at the base of many video editing software tools. For the purpose of testing ffmpeg, v2.6.2 was used to read the video format of the original file and create a re-encoded copy of the file using the ‘–c:v copy’ flag for processing. This flag instructs ffmpeg to not re-encode the video when processing and creates an exact copy of the existing video stream. Comparing the output of an original GoPro video file and a file re-encoded using ffmpeg, shows a clear change in the MPEG-4 structure.



	1	2	3	4	5	6	7	8	9			1	2	3	4	5	6	8	9
1	ftyp										1	ftyp							
2		moov									2	free							
3			mvhd								3	mdat							
4			udta								4	moov							
5				FIRM							5		mvhd						
6				LENS							6		trak						
7				CAME							7			tkhd					
8				SETT							8			edts					
9				AMBA							9				elst				
10				free							10			mdia					
11			trak								11				mdhd				
12				tkhd							12				hdlr				
13				tref							13				minf				
14					tmcd						14					vmhd			
15				edts							15					dinf			
16					elst						16						dref		
17				mdia							17					stbl			
18					mdhd						18						stsd		
19					hdlr						19							avc1	
20					minf						20								avcC
21						vmhd					21						stts		
22						dinf					22						stss		
23							dref				23						stsc		
24						stbl					24						stsz		
25							stsd				25						stco		
26								avc1			26		trak						
27									colr		27			tkhd					
28							stts				28			edts					
29							ctts				29				elst				
30							stsc				30			mdia					
31							stsz				31				mdhd				
32							stco				32				hdlr				
33							stss				33				inf				
34							sdtp				34					smhd			
35			trak								35					dinf			
36				tkhd							36						dref		
37				tref							37					stbl			
38					tmcd						38						stsd		
39				mdia							39							mp4a	
40					mdhd						40								esds
41					hdlr						41						stts		
42					minf						42						stsc		
43						smhd					43						stsz		
44						dinf					44						stco		
45							dref				45		udta						
46						stbl					46			meta					
47							stsd				47				hdlr				
48								mp4a			48				ilst				
49									esds		49				©too				
50							stts				50					data			
51							stsc				51								
52							stsz				52								
53							stco				53								
54			trak								54								
55				tkhd							55								
56				mdia							56								
57					mdhd						57								
58					hdlr						58								
59					minf						59								
60						gmhd					60								
61						hdlr					61								
62						dinf					62								
63							dref				63								
64						stbl					64								
65							stsd				65								
66								tmcd			66								
67							stts				67								
68							stsc				68								
69							stsz				69								
70							stco				70								
71	free										71								
72	mdat										72								

Figure 49. Comparison of Original GoPro Hero 3 and ffmpeg Encoded File Structure

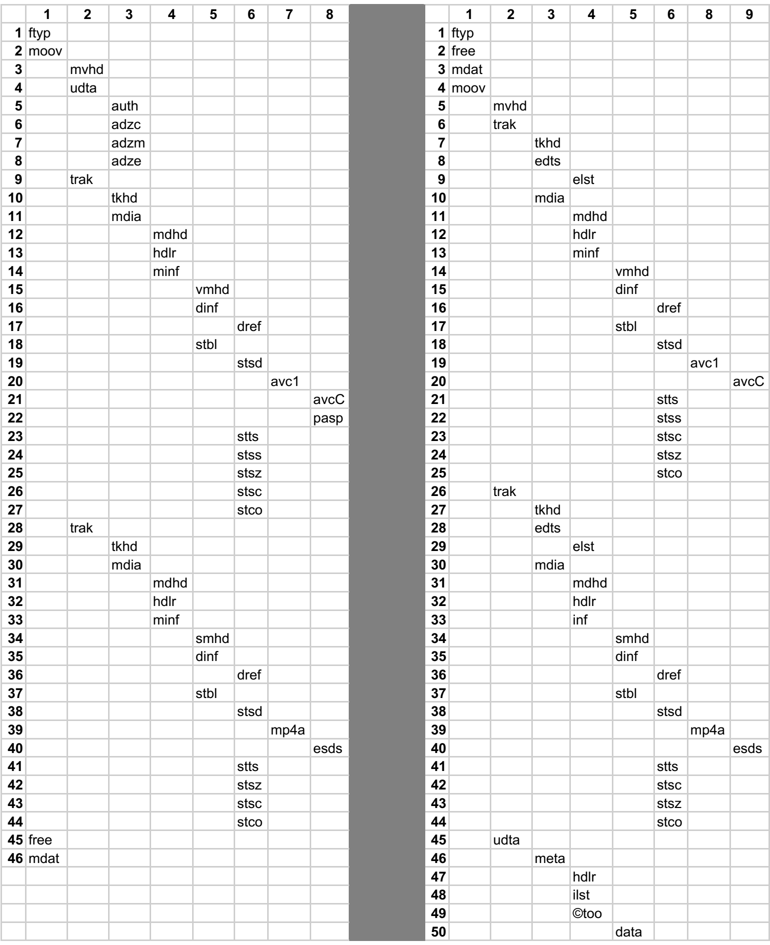
The changes to the structure of the ffmpeg encoded file are distinct and unmistakable. All of the forensically significant user data present in the original GoPro file has been stripped away and when the re-encoded file is further analyzed with MediaInfo, many other changes to the properties of the edited file can be observed. The format profile and codec have changed from ‘JVT’ (Joint Video Team) and ‘avc1’ to ‘Base Media’ and ‘isom’. ffmpeg also zeroes out the embedded timestamps which are reported as the epoch time of January 1, 1904. Among the other changes to the properties of the re-encoded file, another notable addition is the string “Lavf56.25.101” MediaInfo reports as the Writing Application and is contained in the User Data Box (‘udta’) located at the end of the re-encoded file. The string corresponds with the ‘libavformat’ library called by ffmpeg therefore it would be possible to further determine which version of ffmpeg was used for encoding.



General		**General**	
Complete name	1920x1080-GOPRO-HERO3-GOPR1682-BL.mp4	Complete name	gopro_ffmpeg.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	JVT	Format profile	Base Media
Codec ID	avc1	Codec ID	isom
File size	20.3 MiB	File size	21.3 MiB
Duration	6s 440ms	Duration	7s 202ms
Overall bit rate	26.5 Mbps	Overall bit rate mode	Constant
Encoded date	UTC 2015-04-26 17:56:56	Overall bit rate	24.8 Mbps
Tagged date	UTC 2015-04-26 17:56:56	Encoded date	UTC 1904-01-01 00:00:00
AMBA	□□□	Tagged date	UTC 1904-01-01 00:00:00
		Writing application	Lavf56.25.101
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	Main@L4.2	Format profile	Main@L4.2
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=8	Format settings, GOP	M=1, N=8
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	6s 440ms	Duration	7s 174ms
Bit rate mode	Constant	Bit rate mode	Constant
Bit rate	25.0 Mbps	Bit rate	25.0 Mbps
Width	1 920 pixels	Width	1 920 pixels
Height	1 080 pixels	Height	1 080 pixels
Display aspect ratio	16:09	Display aspect ratio	16:09
Frame rate mode	Constant	Frame rate mode	Constant
Frame rate	59.940 fps	Frame rate	59.940 fps
Color space	YUV	Color space	YUV
Chroma subsampling	4:02:00	Chroma subsampling	4:02:00
Bit depth	8 bits	Bit depth	8 bits
Scan type	Progressive	Scan type	Progressive
Bits/(Pixel*Frame)	0.201	Bits/(Pixel*Frame)	0.201
Stream size	19.1 MiB (94%)	Stream size	21.2 MiB (99%)
Title	GoPro AVC	Language	English
Language	English	Encoded date	UTC 1904-01-01 00:00:00
Encoded date	UTC 2015-04-26 17:56:56	Tagged date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-04-26 17:56:56	Color range	Full
Color range	Full	Color primaries	BT.709
Color primaries	BT.709	Transfer characteristics	BT.709
Transfer characteristics	BT.709	Matrix coefficients	BT.709
Matrix coefficients	BT.709		
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	6s 421ms	Duration	7s 202ms
Bit rate mode	Constant	Duration_LastFrame	-9ms
Bit rate	128 Kbps	Bit rate mode	Constant
Channel(s)	2 channels	Bit rate	128 Kbps
Channel positions	Front: L R	Channel(s)	2 channels
Sampling rate	48.0 KHz	Channel positions	Front: L R
Compression mode	Lossy	Sampling rate	48.0 KHz
Stream size	100 KiB (0%)	Compression mode	Lossy
Title	GoPro AAC	Stream size	113 KiB (1%)
Language	English	Language	English
Encoded date	UTC 2015-04-26 17:56:56	Encoded date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-04-26 17:56:56	Tagged date	UTC 1904-01-01 00:00:00
Other			
ID	3		
Type	Time code		
Format	QuickTime TC		
Duration	6s 440ms		
Time code of first frame	17:55:51:27		
Time code, striped	Yes		
Language	English		
Encoded date	UTC 2015-04-26 17:56:56		
Tagged date	UTC 2015-04-26 17:56:56		

Figure 50. MediaInfo Comparison of Original GoPro Hero 3 and ffmpeg Encoded File

When comparing an original file from the LG G3 to the same file that was re-encoded using ffmpeg, the file structure is again distinctly different from the original. The encoding structure of ffmpeg is also consistent with the re-encoding of the GoPro file.

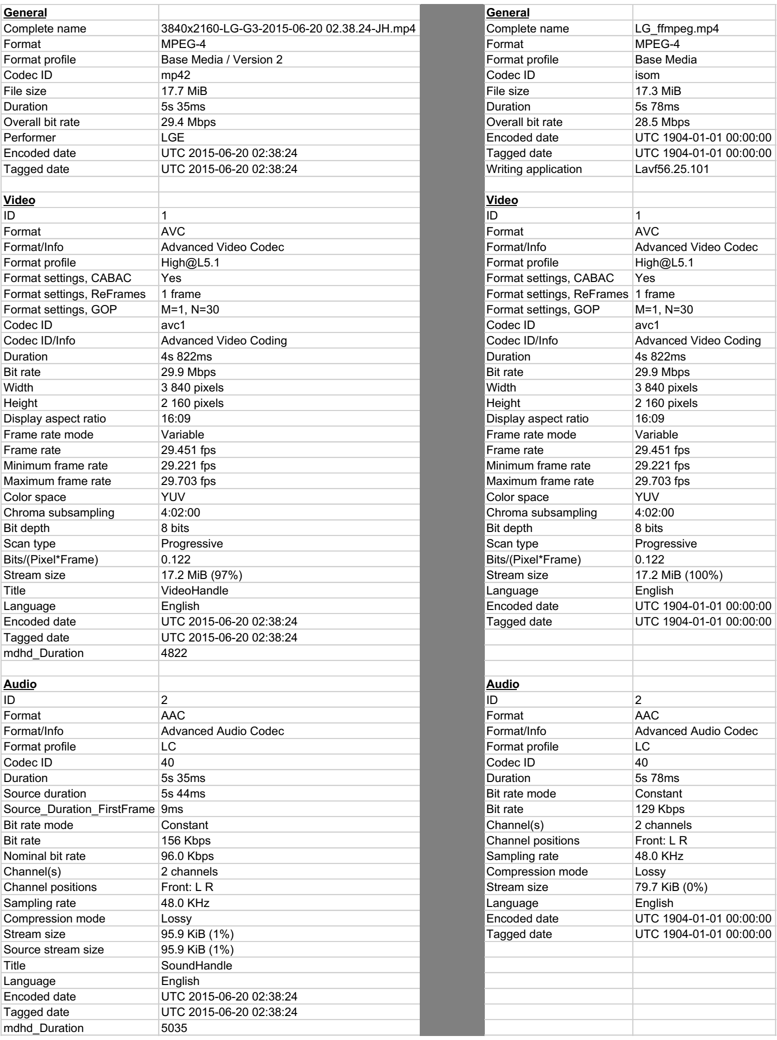


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	free							
3	mdat							
4	moov							
5		mvhd						
6		trak						
7			tkhd					
8			edts					
9				elst				
10			mdia					
11				mdhd				
12				hdlr				
13				minf				
14					vmhd			
15					dinf			
16						dref		
17					stbl			
18						stsd		
19							avc1	
20								avcC
21						stts		
22						stss		
23						stsc		
24						stsz		
25						stco		
26		trak						
27			tkhd					
28			edts					
29				elst				
30			mdia					
31				mdhd				
32				hdlr				
33				inf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsc		
43						stsz		
44						stco		
45		udta						
46			meta					
47				hdlr				
48				ilst				
49				©too				
50					data			

Figure 51. Comparison of LG G3 Original and ffmpeg Encoded File Structure

MediaInfo reports the same series of changes to the properties in the re-encoded LG G3 file as it did with the re-encoded GoPro sample 'file format profile' and 'codec ID' have been modified, the embedded timestamps have been zeroed out, and any identifying metadata has been stripped out and replaced with the same reference to "Lavf56.25.101".

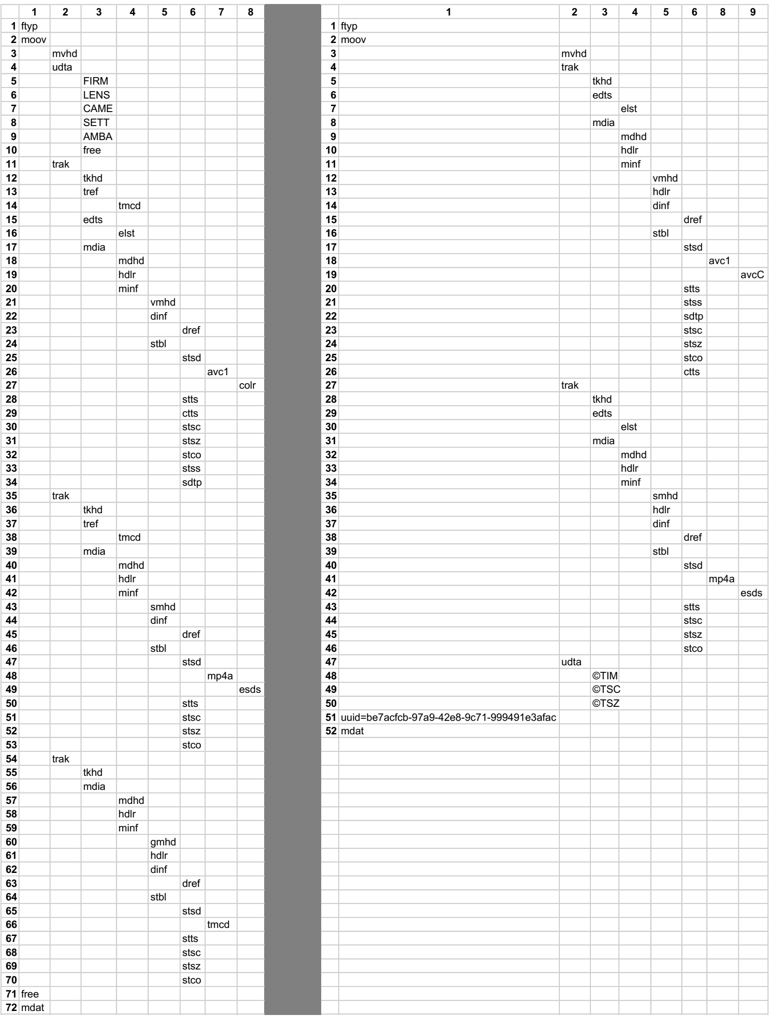


General		**General**	
Complete name	3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4	Complete name	LG_ffmpeg.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	Base Media / Version 2	Format profile	Base Media
Codec ID	mp42	Codec ID	isom
File size	17.7 MiB	File size	17.3 MiB
Duration	5s 35ms	Duration	5s 78ms
Overall bit rate	29.4 Mbps	Overall bit rate	28.5 Mbps
Performer	LGE	Encoded date	UTC 1904-01-01 00:00:00
Encoded date	UTC 2015-06-20 02:38:24	Tagged date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-06-20 02:38:24	Writing application	Lavf56.25.101
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	High@L5.1	Format profile	High@L5.1
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30	Format settings, GOP	M=1, N=30
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	4s 822ms	Duration	4s 822ms
Bit rate	29.9 Mbps	Bit rate	29.9 Mbps
Width	3 840 pixels	Width	3 840 pixels
Height	2 160 pixels	Height	2 160 pixels
Display aspect ratio	16:09	Display aspect ratio	16:09
Frame rate mode	Variable	Frame rate mode	Variable
Frame rate	29.451 fps	Frame rate	29.451 fps
Minimum frame rate	29.221 fps	Minimum frame rate	29.221 fps
Maximum frame rate	29.703 fps	Maximum frame rate	29.703 fps
Color space	YUV	Color space	YUV
Chroma subsampling	4:02:00	Chroma subsampling	4:02:00
Bit depth	8 bits	Bit depth	8 bits
Scan type	Progressive	Scan type	Progressive
Bits/(Pixel*Frame)	0.122	Bits/(Pixel*Frame)	0.122
Stream size	17.2 MiB (97%)	Stream size	17.2 MiB (100%)
Title	VideoHandle	Language	English
Language	English	Encoded date	UTC 1904-01-01 00:00:00
Encoded date	UTC 2015-06-20 02:38:24	Tagged date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-06-20 02:38:24		
mdhd_Duration	4822		
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	5s 35ms	Duration	5s 78ms
Source duration	5s 44ms	Bit rate mode	Constant
Source_Duration_FirstFrame	9ms	Bit rate	129 Kbps
Bit rate mode	Constant	Channel(s)	2 channels
Bit rate	156 Kbps	Channel positions	Front: L R
Nominal bit rate	96.0 Kbps	Sampling rate	48.0 KHz
Channel(s)	2 channels	Compression mode	Lossy
Channel positions	Front: L R	Stream size	79.7 KiB (0%)
Sampling rate	48.0 KHz	Language	English
Compression mode	Lossy	Encoded date	UTC 1904-01-01 00:00:00
Stream size	95.9 KiB (1%)	Tagged date	UTC 1904-01-01 00:00:00
Source stream size	95.9 KiB (1%)		
Title	SoundHandle		
Language	English		
Encoded date	UTC 2015-06-20 02:38:24		
Tagged date	UTC 2015-06-20 02:38:24		
mdhd_Duration	5035		

Figure 52. MediaInfo Comparison of Original LG G3 and ffmpeg Encoded File

Adobe Premiere

Example files were tested against re-encoded versions created with Adobe Premiere CC 2015. Files were imported into Premiere and then exported directly back out using the MPEG-4 settings in the software dialog being careful to match encoder settings without creating any edits in the timeline of the videos themselves. An analysis of the file structure reveals a clear difference between the original GoPro recording and the re-encoded file. The User Data Box (‘udta’) containing the device serial number has been moved within the structure of the file and modified to contain data from Adobe but not from the original file. Adobe inserts a UUID, as well, but it does not appear to be unique to the file itself.

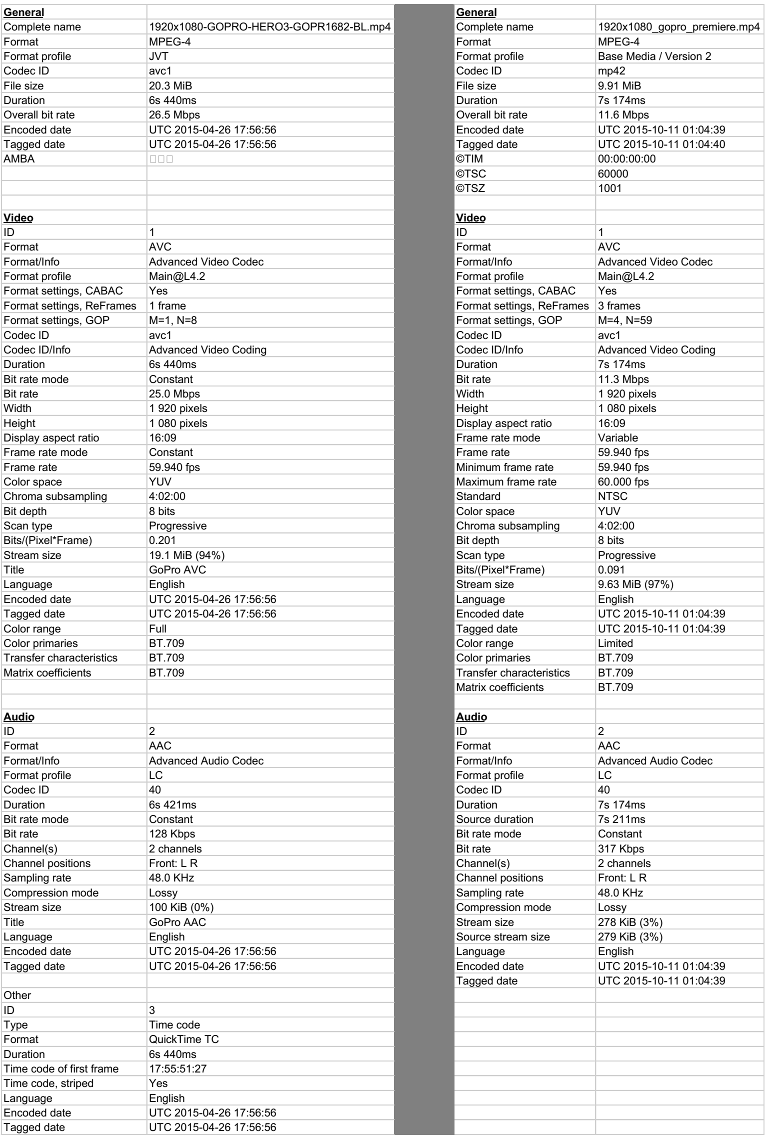


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			FIRM					
6			LENS					
7			CAME					
8			SETT					
9			AMBA					
10			free					
11		trak						
12			tkhd					
13			tref					
14				tmcd				
15			edts					
16				elst				
17			mdia					
18				mdhd				
19				hdlr				
20				minf				
21					vmhd			
22					dinf			
23						dref		
24					stbl			
25						stsd		
26							avc1	
27								colr
28						stts		
29						ctts		
30						stsc		
31						stsz		
32						stco		
33						stss		
34						sdtp		
35		trak						
36			tkhd					
37			tref					
38				tmcd				
39			mdia					
40				mdhd				
41				hdlr				
42				minf				
43					smhd			
44					dinf			
45						dref		
46					stbl			
47						stsd		
48							mp4a	
49								esds
50						stts		
51						stsc		
52						stsz		
53						stco		
54		trak						
55			tkhd					
56			mdia					
57				mdhd				
58				hdlr				
59				minf				
60					gmhd			
61					hdlr			
62					dinf			
63						dref		
64					stbl			
65						stsd		
66							tmcd	
67						stts		
68						stsc		
69						stsz		
70						stco		
71	free							
72	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	moov							
3		mvhd						
4		trak						
5			tkhd					
6			edts					
7				elst				
8			mdia					
9				mdhd				
10				hdlr				
11				minf				
12					vmhd			
13					hdlr			
14					dinf			
15						dref		
16					stbl			
17						stsd		
18							avc1	
19								avcC
20						stts		
21						stss		
22						sdtp		
23						stsc		
24						stsz		
25						stco		
26						ctts		
27		trak						
28			tkhd					
29			edts					
30				elst				
31			mdia					
32				mdhd				
33				hdlr				
34				minf				
35					smhd			
36					hdlr			
37					dinf			
38						dref		
39					stbl			
40						stsd		
41							mp4a	
42								esds
43						stts		
44						stsc		
45						stsz		
46						stco		
47		udta						
48			©TIM					
49			©TSC					
50			©TSZ					
51	uuid=be7acfcb-97a9-42e8-9c71-999491e3afac							
52	mdat							

Figure 53. Comparison of GoPro Hero 3 Original and Adobe Premiere Encoded File Structure

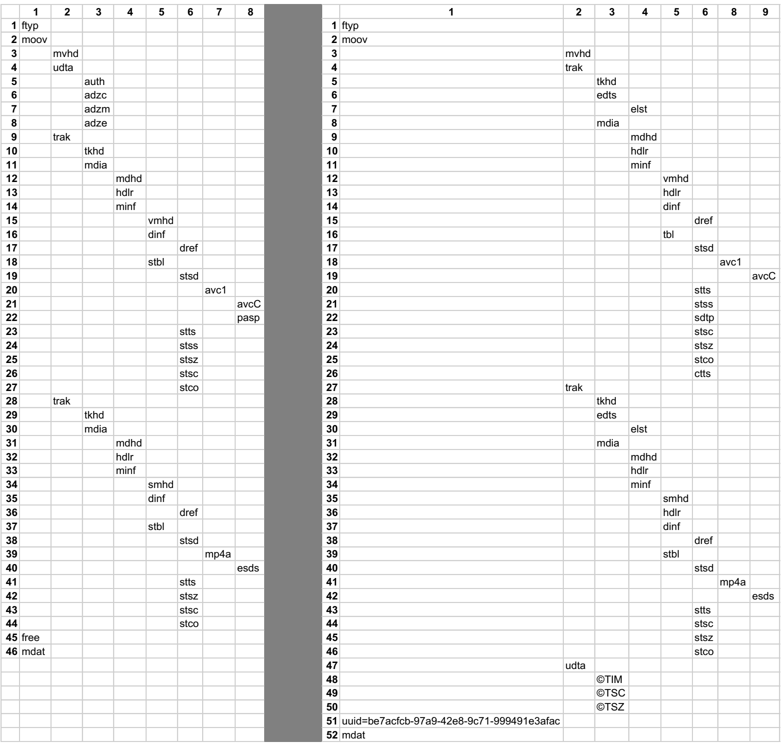
An analysis with MediaInfo reveals that the format profile and codec ID have been modified by Adobe Premiere. The embedded timestamps have been updated from the original time to the time of the re-encoding. There are other changes to the properties of the re-encoded file but most notable is the absence of the QuickTime Time Code track contained in the original GoPro file.



General		**General**	
Complete name	1920x1080-GOPRO-HERO3-GOPR1682-BL.mp4	Complete name	1920x1080_gopro_premiere.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	JVT	Format profile	Base Media / Version 2
Codec ID	avc1	Codec ID	mp42
File size	20.3 MiB	File size	9.91 MiB
Duration	6s 440ms	Duration	7s 174ms
Overall bit rate	26.5 Mbps	Overall bit rate	11.6 Mbps
Encoded date	UTC 2015-04-26 17:56:56	Encoded date	UTC 2015-10-11 01:04:39
Tagged date	UTC 2015-04-26 17:56:56	Tagged date	UTC 2015-10-11 01:04:40
AMBA	□□□	©TIM	00:00:00:00
		©TSC	60000
		©TSZ	1001
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	Main@L4.2	Format profile	Main@L4.2
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	3 frames
Format settings, GOP	M=1, N=8	Format settings, GOP	M=4, N=59
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	6s 440ms	Duration	7s 174ms
Bit rate mode	Constant	Bit rate	11.3 Mbps
Bit rate	25.0 Mbps	Width	1 920 pixels
Width	1 920 pixels	Height	1 080 pixels
Height	1 080 pixels	Display aspect ratio	16:09
Display aspect ratio	16:09	Frame rate mode	Variable
Frame rate mode	Constant	Frame rate	59.940 fps
Frame rate	59.940 fps	Minimum frame rate	59.940 fps
Color space	YUV	Maximum frame rate	60.000 fps
Chroma subsampling	4:02:00	Standard	NTSC
Bit depth	8 bits	Color space	YUV
Scan type	Progressive	Chroma subsampling	4:02:00
Bits/(Pixel*Frame)	0.201	Bit depth	8 bits
Stream size	19.1 MiB (94%)	Scan type	Progressive
Title	GoPro AVC	Bits/(Pixel*Frame)	0.091
Language	English	Stream size	9.63 MiB (97%)
Encoded date	UTC 2015-04-26 17:56:56	Language	English
Tagged date	UTC 2015-04-26 17:56:56	Encoded date	UTC 2015-10-11 01:04:39
Color range	Full	Tagged date	UTC 2015-10-11 01:04:39
Color primaries	BT.709	Color range	Limited
Transfer characteristics	BT.709	Color primaries	BT.709
Matrix coefficients	BT.709	Transfer characteristics	BT.709
		Matrix coefficients	BT.709
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	6s 421ms	Duration	7s 174ms
Bit rate mode	Constant	Source duration	7s 211ms
Bit rate	128 Kbps	Bit rate mode	Constant
Channel(s)	2 channels	Bit rate	317 Kbps
Channel positions	Front: L R	Channel(s)	2 channels
Sampling rate	48.0 KHz	Channel positions	Front: L R
Compression mode	Lossy	Sampling rate	48.0 KHz
Stream size	100 KiB (0%)	Compression mode	Lossy
Title	GoPro AAC	Stream size	278 KiB (3%)
Language	English	Source stream size	279 KiB (3%)
Encoded date	UTC 2015-04-26 17:56:56	Language	English
Tagged date	UTC 2015-04-26 17:56:56	Encoded date	UTC 2015-10-11 01:04:39
		Tagged date	UTC 2015-10-11 01:04:39
Other			
ID	3		
Type	Time code		
Format	QuickTime TC		
Duration	6s 440ms		
Time code of first frame	17:55:51:27		
Time code, striped	Yes		
Language	English		
Encoded date	UTC 2015-04-26 17:56:56		
Tagged date	UTC 2015-04-26 17:56:56		

Figure 54. MediaInfo Comparison of Original GoPro Hero 3 and Adobe Premiere Encoded File

Comparing the original LG G3 recording to the re-encoded copy created with Adobe Premiere shows an identical change to MPEG-4 file structure as was observed with the GoPro re-encoding. The embedded UUID is identical and again any user data in the original file has been stripped away and replaced with Adobe's own content.

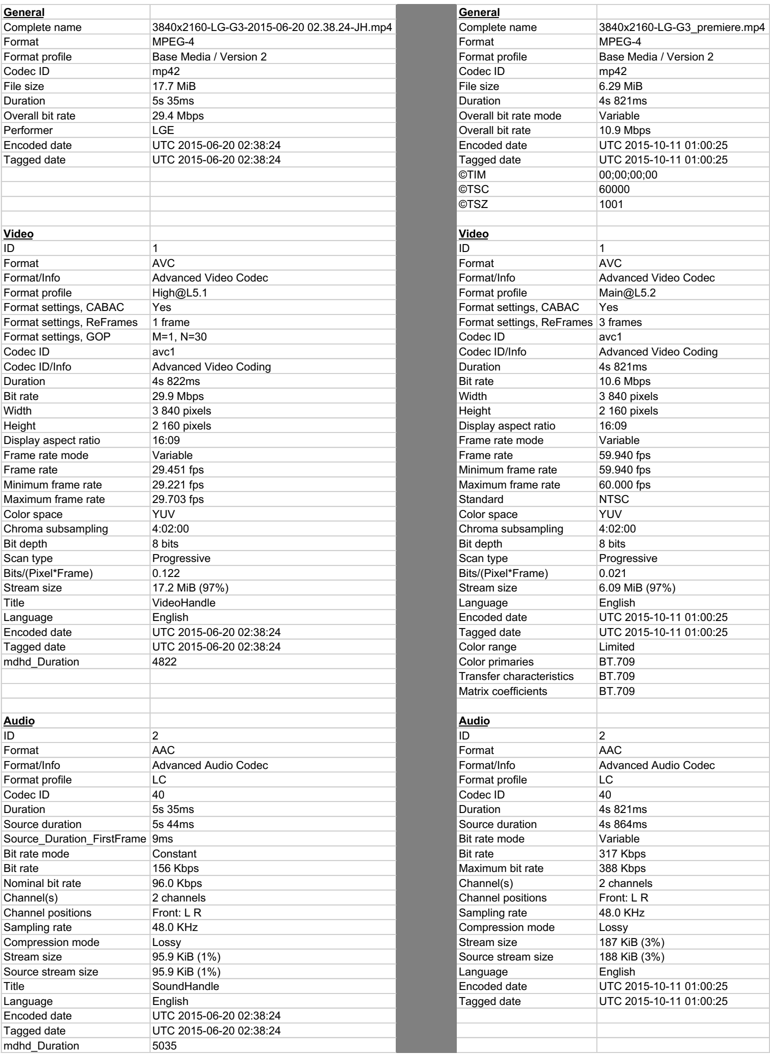


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	moov							
3		mvhd						
4		trak						
5			tkhd					
6			edts					
7				elst				
8			mdia					
9				mdhd				
10				hdlr				
11				minf				
12					vmhd			
13					hdlr			
14					dinf			
15						dref		
16					tbl			
17						stsd		
18							avc1	
19								avcC
20						stts		
21						stss		
22						sdtp		
23						stsc		
24						stsz		
25						stco		
26						ctts		
27		trak						
28			tkhd					
29			edts					
30				elst				
31			mdia					
32				mdhd				
33				hdlr				
34				minf				
35					smhd			
36					hdlr			
37					dinf			
38						dref		
39					stbl			
40						stsd		
41							mp4a	
42								esds
43						stts		
44						stsc		
45						stsz		
46						stco		
47		udta						
48			©TIM					
49			©TSC					
50			©TSZ					
51	uuid=be7acfcb-97a9-42e8-9c71-999491e3afac							
52	mdat							

Figure 55. Comparison of Original LG G3 and Adobe Premiere Encoded File Structure

An analysis with MediaInfo reveals the change expected to the embedded timestamps but a file recorded at 60fps rather than at the 30fps of the original. There are other inclusions and exclusions in the properties of the re-encoded file and this level of analysis will only serve to confirm or deny a match between files. However, at the most basic level a keyword search of either file created by Adobe Premiere reveals fifteen hits for the string ‘adobe’ in the metadata of the file itself.

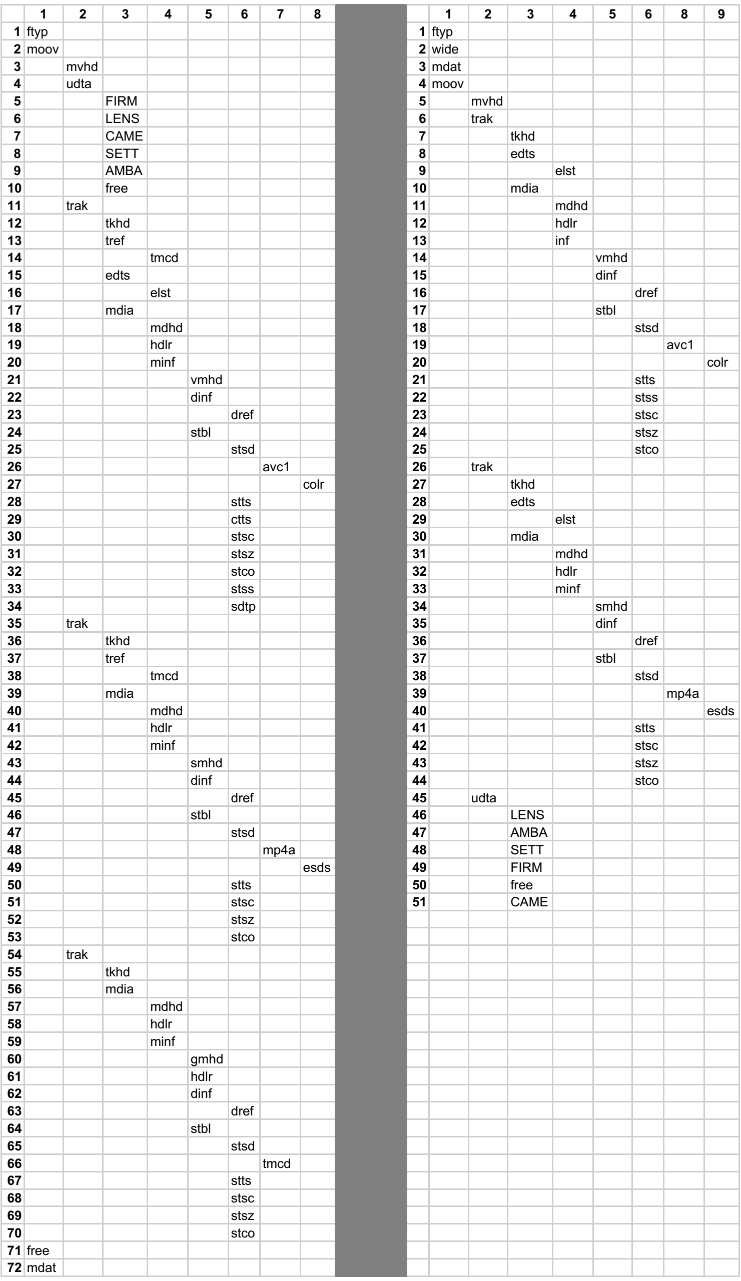


General		**General**	
Complete name	3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4	Complete name	3840x2160-LG-G3_premiere.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	Base Media / Version 2	Format profile	Base Media / Version 2
Codec ID	mp42	Codec ID	mp42
File size	17.7 MiB	File size	6.29 MiB
Duration	5s 35ms	Duration	4s 821ms
Overall bit rate	29.4 Mbps	Overall bit rate mode	Variable
Performer	LGE	Overall bit rate	10.9 Mbps
Encoded date	UTC 2015-06-20 02:38:24	Encoded date	UTC 2015-10-11 01:00:25
Tagged date	UTC 2015-06-20 02:38:24	Tagged date	UTC 2015-10-11 01:00:25
		©TIM	00;00;00;00
		©TSC	60000
		©TSZ	1001
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	High@L5.1	Format profile	Main@L5.2
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	3 frames
Format settings, GOP	M=1, N=30	Codec ID	avc1
Codec ID	avc1	Codec ID/Info	Advanced Video Coding
Codec ID/Info	Advanced Video Coding	Duration	4s 821ms
Duration	4s 822ms	Bit rate	10.6 Mbps
Bit rate	29.9 Mbps	Width	3 840 pixels
Width	3 840 pixels	Height	2 160 pixels
Height	2 160 pixels	Display aspect ratio	16:09
Display aspect ratio	16:09	Frame rate mode	Variable
Frame rate mode	Variable	Frame rate	59.940 fps
Frame rate	29.451 fps	Minimum frame rate	59.940 fps
Minimum frame rate	29.221 fps	Maximum frame rate	60.000 fps
Maximum frame rate	29.703 fps	Standard	NTSC
Color space	YUV	Color space	YUV
Chroma subsampling	4:02:00	Chroma subsampling	4:02:00
Bit depth	8 bits	Bit depth	8 bits
Scan type	Progressive	Scan type	Progressive
Bits/(Pixel*Frame)	0.122	Bits/(Pixel*Frame)	0.021
Stream size	17.2 MiB (97%)	Stream size	6.09 MiB (97%)
Title	VideoHandle	Language	English
Language	English	Encoded date	UTC 2015-10-11 01:00:25
Encoded date	UTC 2015-06-20 02:38:24	Tagged date	UTC 2015-10-11 01:00:25
Tagged date	UTC 2015-06-20 02:38:24	Color range	Limited
mdhd_Duration	4822	Color primaries	BT.709
		Transfer characteristics	BT.709
		Matrix coefficients	BT.709
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	5s 35ms	Duration	4s 821ms
Source duration	5s 44ms	Source duration	4s 864ms
Source_Duration_FirstFrame	9ms	Bit rate mode	Variable
Bit rate mode	Constant	Bit rate	317 Kbps
Bit rate	156 Kbps	Maximum bit rate	388 Kbps
Nominal bit rate	96.0 Kbps	Channel(s)	2 channels
Channel(s)	2 channels	Channel positions	Front: L R
Channel positions	Front: L R	Sampling rate	48.0 KHz
Sampling rate	48.0 KHz	Compression mode	Lossy
Compression mode	Lossy	Stream size	187 KiB (3%)
Stream size	95.9 KiB (1%)	Source stream size	188 KiB (3%)
Source stream size	95.9 KiB (1%)	Language	English
Title	SoundHandle	Encoded date	UTC 2015-10-11 01:00:25
Language	English	Tagged date	UTC 2015-10-11 01:00:25
Encoded date	UTC 2015-06-20 02:38:24		
Tagged date	UTC 2015-06-20 02:38:24		
mdhd_Duration	5035		

Figure 56. MediaInfo Comparison of Original LG G3 and Adobe Premiere Encoded File

Apple Quicktime

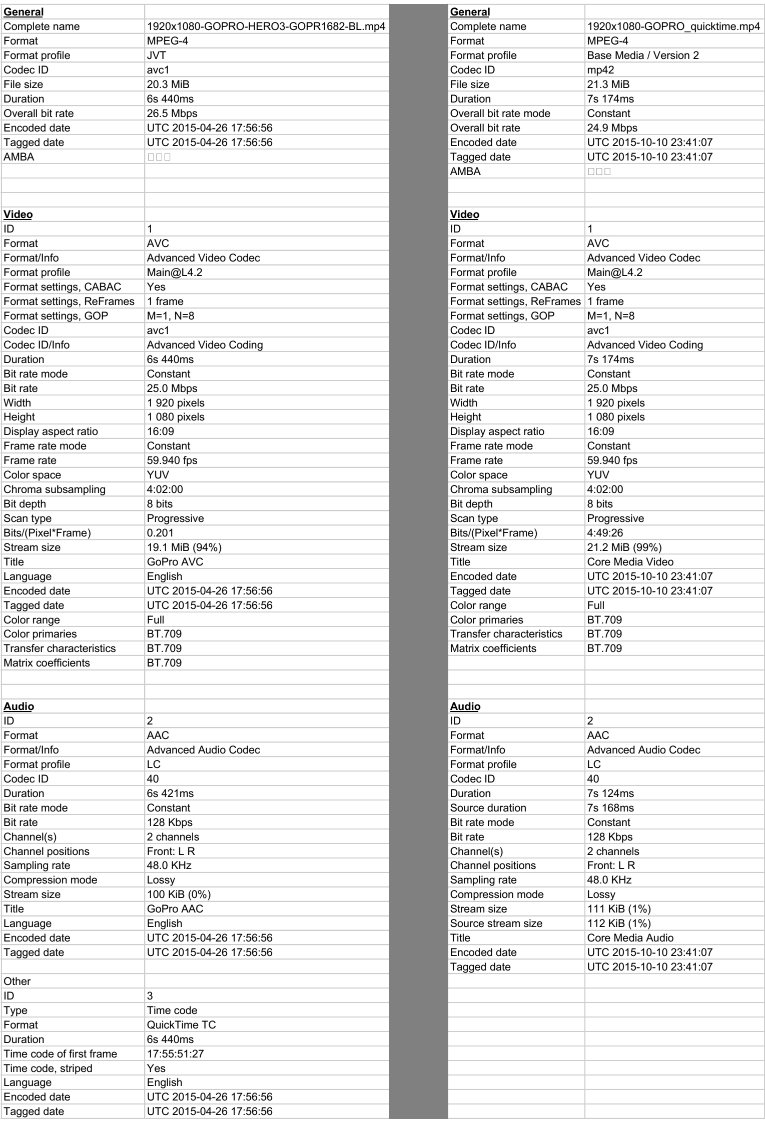
To test another encoding engine, Apple's QuickTime Player v.10.4 was used to re-encode the sample files for analysis and comparison using its Export function to re-encode the two sample files being examined. The MPEG-4 structure of a file re-encoded with QuickTime shows clear differences from the original GoPro recording. The QuickTime Time Code track has been stripped away but it should be noted that QuickTime is the first piece of software to make any attempt to preserve the contents of the User Data Box ('udta') present in the original file. To verify the preservation of the User Data Box ('udta') contents between the original and the re-encoded file, these boxes were examined separately to confirm their data. QuickTime has re-arranged these boxes but their contents remain valid.



	1	2	3	4	5	6	7	8			1	2	3	4	5	6	8	9
1	ftyp									1	ftyp							
2	moov									2	wide							
3		mvhd								3	mdat							
4		udta								4	moov							
5			FIRM							5		mvhd						
6			LENS							6		trak						
7			CAME							7			tkhd					
8			SETT							8			edts					
9			AMBA							9				elst				
10			free							10			mdia					
11		trak								11				mdhd				
12			tkhd							12				hdlr				
13			tref							13				inf				
14				tmcd						14					vmhd			
15			edts							15					dinf			
16				elst						16						dref		
17			mdia							17					stbl			
18				mdhd						18						stsd		
19				hdlr						19							avc1	
20				minf						20								colr
21					vmhd					21						stts		
22					dinf					22						stss		
23						dref				23						stsc		
24					stbl					24						stsz		
25						stsd				25						stco		
26							avc1			26		trak						
27								colr		27			tkhd					
28						stts				28			edts					
29						ctts				29				elst				
30						stsc				30			mdia					
31						stsz				31				mdhd				
32						stco				32				hdlr				
33						stss				33				minf				
34						sdtp				34					smhd			
35		trak								35					dinf			
36			tkhd							36						dref		
37			tref							37					stbl			
38				tmcd						38						stsd		
39			mdia							39							mp4a	
40				mdhd						40								esds
41				hdlr						41						stts		
42				minf						42						stsc		
43					smhd					43						stsz		
44					dinf					44						stco		
45						dref				45		udta						
46					stbl					46			LENS					
47						stsd				47			AMBA					
48							mp4a			48			SETT					
49								esds		49			FIRM					
50						stts				50			free					
51						stsc				51			CAME					
52						stsz												
53						stco												
54		trak																
55			tkhd															
56			mdia															
57				mdhd														
58				hdlr														
59				minf														
60					gmhd													
61					hdlr													
62					dinf													
63						dref												
64					stbl													
65						stsd												
66							tmcd											
67						stts												
68						stsc												
69						stsz												
70						stco												
71	free																	
72	mdat																	

Figure 57. Comparison of GoPro Hero 3 Original and Apple QuickTime Encoded File Structure

Examining the file with MediaInfo shows that the format profile and the codec ID have changed, the embedded timestamps have been updated to the time of re-encoding, and two pieces of self-identifying GoPro references have been stripped away from the audio and video tracks.

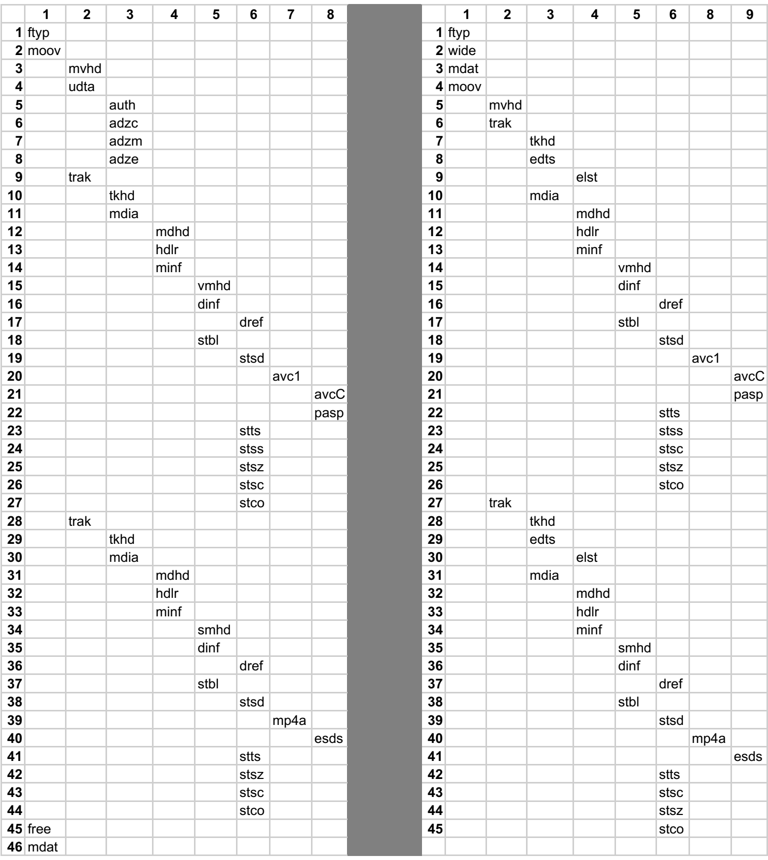


General		**General**	
Complete name	1920x1080-GOPRO-HERO3-GOPR1682-BL.mp4	Complete name	1920x1080-GOPRO_quicktime.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	JVT	Format profile	Base Media / Version 2
Codec ID	avc1	Codec ID	mp42
File size	20.3 MiB	File size	21.3 MiB
Duration	6s 440ms	Duration	7s 174ms
Overall bit rate	26.5 Mbps	Overall bit rate mode	Constant
Encoded date	UTC 2015-04-26 17:56:56	Overall bit rate	24.9 Mbps
Tagged date	UTC 2015-04-26 17:56:56	Encoded date	UTC 2015-10-10 23:41:07
AMBA	□□□	Tagged date	UTC 2015-10-10 23:41:07
		AMBA	□□□
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	Main@L4.2	Format profile	Main@L4.2
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=8	Format settings, GOP	M=1, N=8
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	6s 440ms	Duration	7s 174ms
Bit rate mode	Constant	Bit rate mode	Constant
Bit rate	25.0 Mbps	Bit rate	25.0 Mbps
Width	1 920 pixels	Width	1 920 pixels
Height	1 080 pixels	Height	1 080 pixels
Display aspect ratio	16:09	Display aspect ratio	16:09
Frame rate mode	Constant	Frame rate mode	Constant
Frame rate	59.940 fps	Frame rate	59.940 fps
Color space	YUV	Color space	YUV
Chroma subsampling	4:02:00	Chroma subsampling	4:02:00
Bit depth	8 bits	Bit depth	8 bits
Scan type	Progressive	Scan type	Progressive
Bits/(Pixel*Frame)	0.201	Bits/(Pixel*Frame)	4:49:26
Stream size	19.1 MiB (94%)	Stream size	21.2 MiB (99%)
Title	GoPro AVC	Title	Core Media Video
Language	English	Encoded date	UTC 2015-10-10 23:41:07
Encoded date	UTC 2015-04-26 17:56:56	Tagged date	UTC 2015-10-10 23:41:07
Tagged date	UTC 2015-04-26 17:56:56	Color range	Full
Color range	Full	Color primaries	BT.709
Color primaries	BT.709	Transfer characteristics	BT.709
Transfer characteristics	BT.709	Matrix coefficients	BT.709
Matrix coefficients	BT.709		
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	6s 421ms	Duration	7s 124ms
Bit rate mode	Constant	Source duration	7s 168ms
Bit rate	128 Kbps	Bit rate mode	Constant
Channel(s)	2 channels	Bit rate	128 Kbps
Channel positions	Front: L R	Channel(s)	2 channels
Sampling rate	48.0 KHz	Channel positions	Front: L R
Compression mode	Lossy	Sampling rate	48.0 KHz
Stream size	100 KiB (0%)	Compression mode	Lossy
Title	GoPro AAC	Stream size	111 KiB (1%)
Language	English	Source stream size	112 KiB (1%)
Encoded date	UTC 2015-04-26 17:56:56	Title	Core Media Audio
Tagged date	UTC 2015-04-26 17:56:56	Encoded date	UTC 2015-10-10 23:41:07
		Tagged date	UTC 2015-10-10 23:41:07
Other			
ID	3		
Type	Time code		
Format	QuickTime TC		
Duration	6s 440ms		
Time code of first frame	17:55:51:27		
Time code, striped	Yes		
Language	English		
Encoded date	UTC 2015-04-26 17:56:56		
Tagged date	UTC 2015-04-26 17:56:56		

Figure 58. MediaInfo Comparison of GoPro Hero 3 Original and Apple QuickTime Encoded File

Using AtomicParsley to compare the structures of the original LG G3 file and the QuickTime re-encoded file shows distinct differences in the MPEG-4 structure that would allow the QuickTime file to be identified as being not original.

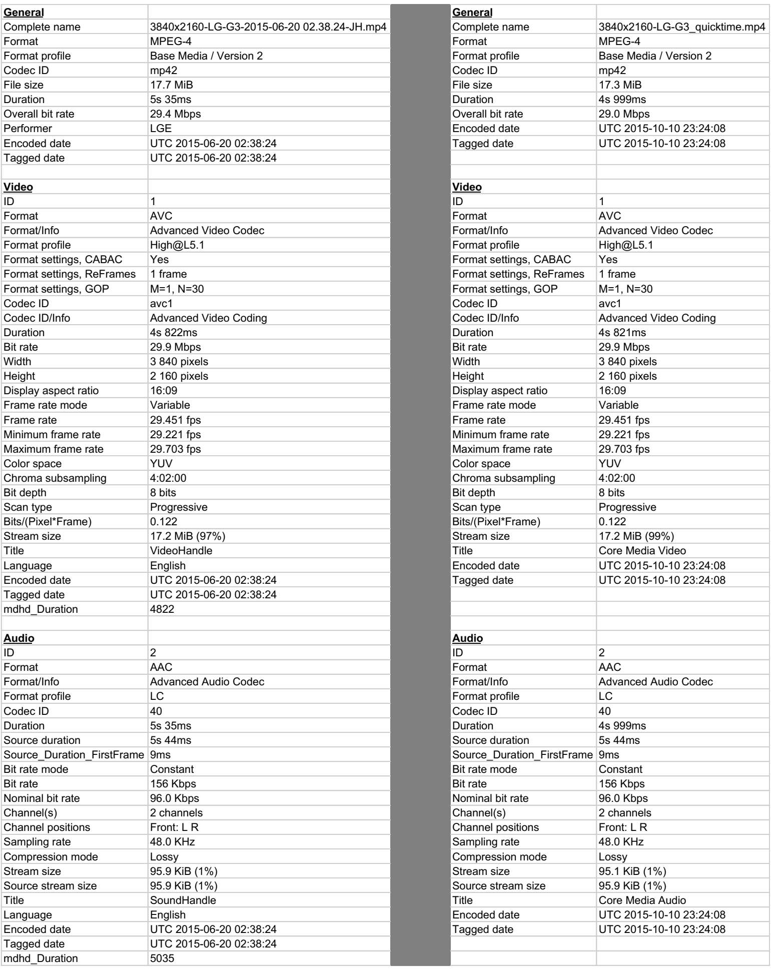
That being said, the structure of the re-encoded LG G3 file is not the same as the structure of the re-encoded GoPro file. It seems that QuickTime takes certain parts of the original file's structure into account when re-encoding rather than re-encoding using a strict structure as observed with ffmpeg and Adobe Premiere. While there was no meaningful data contained in the User Data Box ('udta') of the original file this data was not preserved during re-encoding as it was in the case of the GoPro.



	1	2	3	4	5	6	7	8			1	2	3	4	5	6	8	9
1	ftyp									1	ftyp							
2	moov									2	wide							
3		mvhd								3	mdat							
4		udta								4	moov							
5			auth							5		mvhd						
6			adzc							6		trak						
7			adzm							7			tkhd					
8			adze							8			edts					
9		trak								9				elst				
10			tkhd							10			mdia					
11			mdia							11				mdhd				
12				mdhd						12				hdlr				
13				hdlr						13				minf				
14				minf						14					vmhd			
15					vmhd					15					dinf			
16					dinf					16						dref		
17						dref				17					stbl			
18					stbl					18						stsd		
19						stsd				19							avc1	
20							avc1			20								avcC
21								avcC		21								pasp
22								pasp		22						stts		
23						stts				23						stss		
24						stss				24						stsc		
25						stsz				25						stsz		
26						stsc				26						stco		
27						stco				27		trak						
28		trak								28			tkhd					
29			tkhd							29			edts					
30			mdia							30				elst				
31				mdhd						31			mdia					
32				hdlr						32				mdhd				
33				minf						33				hdlr				
34					smhd					34				minf				
35					dinf					35					smhd			
36						dref				36					dinf			
37					stbl					37						dref		
38						stsd				38					stbl			
39							mp4a			39						stsd		
40								esds		40							mp4a	
41						stts				41								esds
42						stsz				42						stts		
43						stsc				43						stsc		
44						stco				44						stsz		
45	free									45						stco		
46	mdat																	

Figure 59. Comparison of LG G3 Original and Apple QuickTime Encoded File Structure

Analysis with MediaInfo shows that the embedded timestamps have been updated to the time of re-encoding, the self-identifying reference 'LGE' has been removed, as well as the references to 'VideoHandle' and 'SoundHandle.'



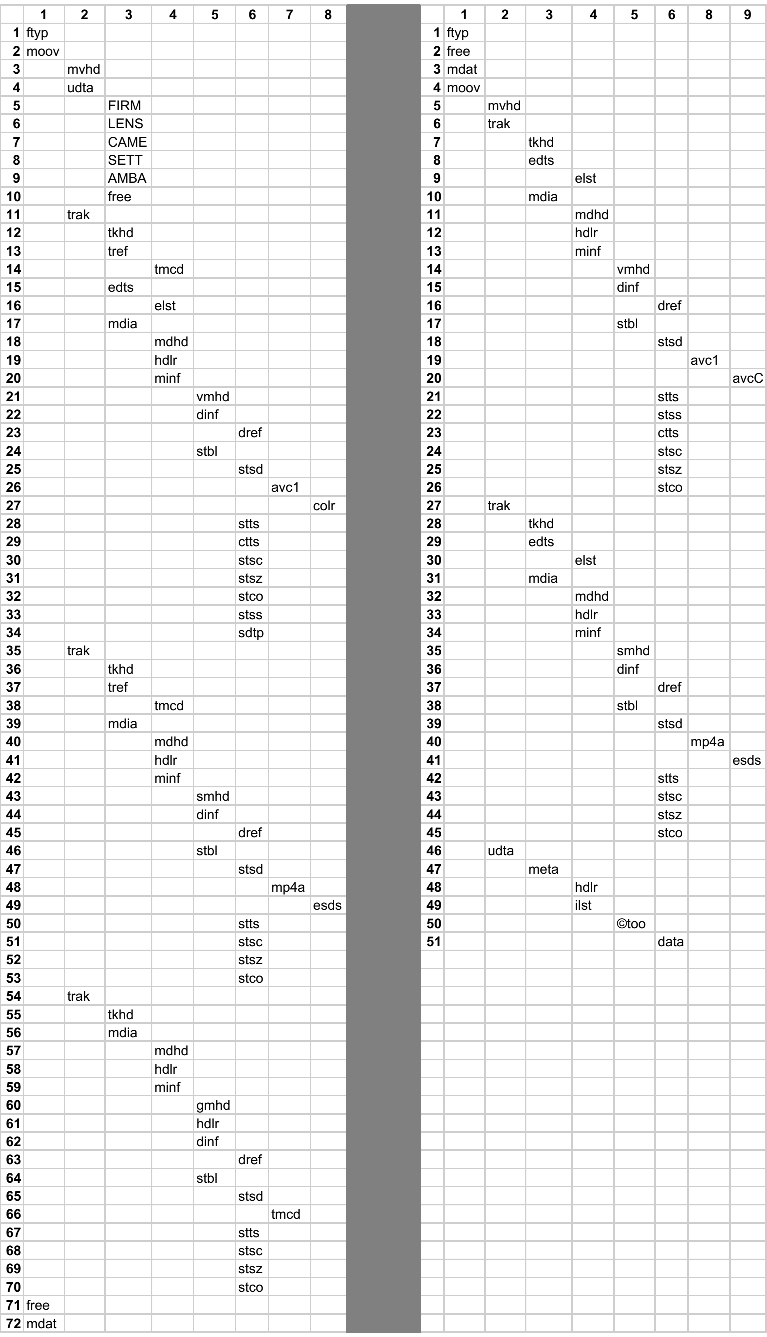
General			**General**	
Complete name	3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4		Complete name	3840x2160-LG-G3_quicktime.mp4
Format	MPEG-4		Format	MPEG-4
Format profile	Base Media / Version 2		Format profile	Base Media / Version 2
Codec ID	mp42		Codec ID	mp42
File size	17.7 MiB		File size	17.3 MiB
Duration	5s 35ms		Duration	4s 999ms
Overall bit rate	29.4 Mbps		Overall bit rate	29.0 Mbps
Performer	LGE		Encoded date	UTC 2015-10-10 23:24:08
Encoded date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-10-10 23:24:08
Tagged date	UTC 2015-06-20 02:38:24			
Video			**Video**	
ID	1		ID	1
Format	AVC		Format	AVC
Format/Info	Advanced Video Codec		Format/Info	Advanced Video Codec
Format profile	High@L5.1		Format profile	High@L5.1
Format settings, CABAC	Yes		Format settings, CABAC	Yes
Format settings, ReFrames	1 frame		Format settings, ReFrames	1 frame
Format settings, GOP	M=1, N=30		Format settings, GOP	M=1, N=30
Codec ID	avc1		Codec ID	avc1
Codec ID/Info	Advanced Video Coding		Codec ID/Info	Advanced Video Coding
Duration	4s 822ms		Duration	4s 821ms
Bit rate	29.9 Mbps		Bit rate	29.9 Mbps
Width	3 840 pixels		Width	3 840 pixels
Height	2 160 pixels		Height	2 160 pixels
Display aspect ratio	16:09		Display aspect ratio	16:09
Frame rate mode	Variable		Frame rate mode	Variable
Frame rate	29.451 fps		Frame rate	29.451 fps
Minimum frame rate	29.221 fps		Minimum frame rate	29.221 fps
Maximum frame rate	29.703 fps		Maximum frame rate	29.703 fps
Color space	YUV		Color space	YUV
Chroma subsampling	4:02:00		Chroma subsampling	4:02:00
Bit depth	8 bits		Bit depth	8 bits
Scan type	Progressive		Scan type	Progressive
Bits/(Pixel*Frame)	0.122		Bits/(Pixel*Frame)	0.122
Stream size	17.2 MiB (97%)		Stream size	17.2 MiB (99%)
Title	VideoHandle		Title	Core Media Video
Language	English		Encoded date	UTC 2015-10-10 23:24:08
Encoded date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-10-10 23:24:08
Tagged date	UTC 2015-06-20 02:38:24			
mdhd_Duration	4822			
Audio			**Audio**	
ID	2		ID	2
Format	AAC		Format	AAC
Format/Info	Advanced Audio Codec		Format/Info	Advanced Audio Codec
Format profile	LC		Format profile	LC
Codec ID	40		Codec ID	40
Duration	5s 35ms		Duration	4s 999ms
Source duration	5s 44ms		Source duration	5s 44ms
Source_Duration_FirstFrame	9ms		Source_Duration_FirstFrame	9ms
Bit rate mode	Constant		Bit rate mode	Constant
Bit rate	156 Kbps		Bit rate	156 Kbps
Nominal bit rate	96.0 Kbps		Nominal bit rate	96.0 Kbps
Channel(s)	2 channels		Channel(s)	2 channels
Channel positions	Front: L R		Channel positions	Front: L R
Sampling rate	48.0 KHz		Sampling rate	48.0 KHz
Compression mode	Lossy		Compression mode	Lossy
Stream size	95.9 KiB (1%)		Stream size	95.1 KiB (1%)
Source stream size	95.9 KiB (1%)		Source stream size	95.9 KiB (1%)
Title	SoundHandle		Title	Core Media Audio
Language	English		Encoded date	UTC 2015-10-10 23:24:08
Encoded date	UTC 2015-06-20 02:38:24		Tagged date	UTC 2015-10-10 23:24:08
Tagged date	UTC 2015-06-20 02:38:24			
mdhd_Duration	5035			

Figure 60. MediaInfo Comparison of LG G3 Original and Apple QuickTime Encoded File

youtube-dl

As a final test of the methods of analysis outlined in this paper, the sample clips from the GoPro Hero 3 and LG G3 were uploaded to YouTube and then downloaded using ‘youtube-dl’ version 2015.10.09. This software is released into the public domain and is available online at https://github.com/rg3/youtube-dl/ These downloaded files were then compared with the original files in order to compare the files created by a popular tool used for downloading YouTube videos.

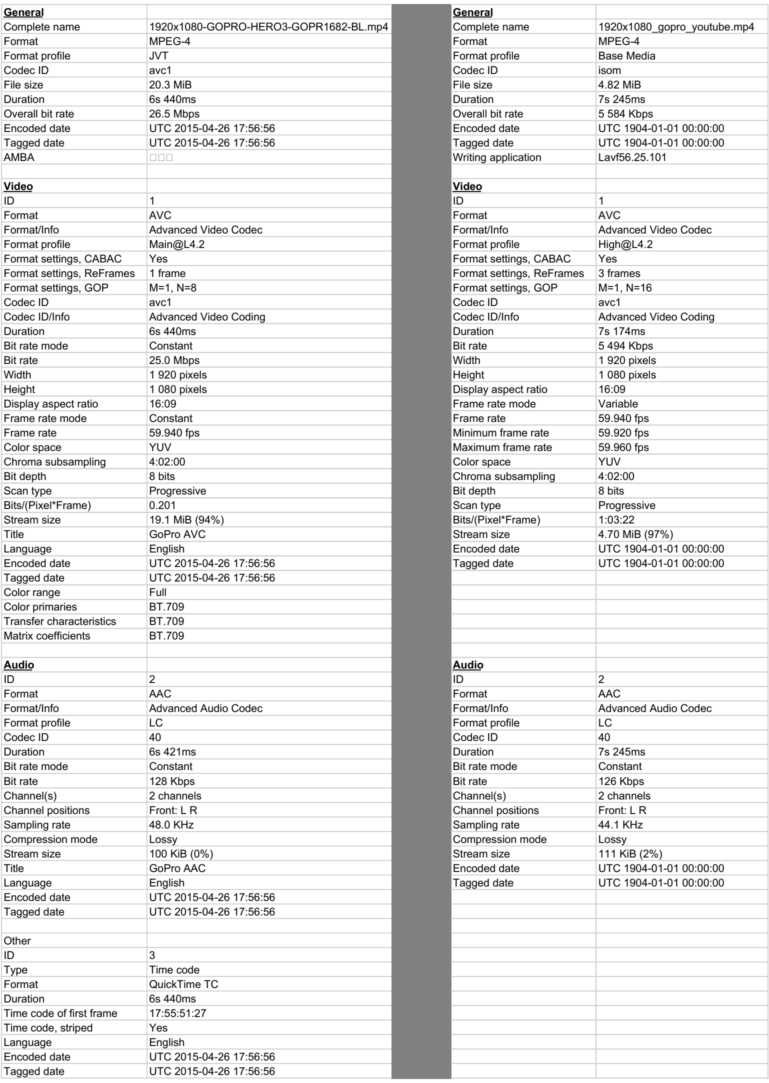
Using AtomicParsley to extract the file structure of the YouTube re-encoded file reveals a file structure very different from the original and appears to be the same output structure as was observed in the ffmpeg structure analysis.



	1	2	3	4	5	6	7	8			1	2	3	4	5	6	8	9
1	ftyp									1	ftyp							
2	moov									2	free							
3		mvhd								3	mdat							
4		udta								4	moov							
5			FIRM							5		mvhd						
6			LENS							6		trak						
7			CAME							7			tkhd					
8			SETT							8			edts					
9			AMBA							9				elst				
10			free							10			mdia					
11		trak								11				mdhd				
12			tkhd							12				hdlr				
13			tref							13				minf				
14				tmcd						14					vmhd			
15			edts							15					dinf			
16				elst						16						dref		
17			mdia							17					stbl			
18				mdhd						18						stsd		
19				hdlr						19							avc1	
20				minf						20								avcC
21					vmhd					21						stts		
22					dinf					22						stss		
23						dref				23						ctts		
24					stbl					24						stsc		
25						stsd				25						stsz		
26							avc1			26						stco		
27								colr		27		trak						
28						stts				28			tkhd					
29						ctts				29			edts					
30						stsc				30				elst				
31						stsz				31			mdia					
32						stco				32				mdhd				
33						stss				33				hdlr				
34						sdtp				34				minf				
35		trak								35					smhd			
36			tkhd							36					dinf			
37			tref							37						dref		
38				tmcd						38					stbl			
39			mdia							39						stsd		
40				mdhd						40							mp4a	
41				hdlr						41								esds
42				minf						42						stts		
43					smhd					43						stsc		
44					dinf					44						stsz		
45						dref				45						stco		
46					stbl					46		udta						
47						stsd				47			meta					
48							mp4a			48				hdlr				
49								esds		49				ilst				
50						stts				50					©too			
51						stsc				51						data		
52						stsz												
53						stco												
54		trak																
55			tkhd															
56			mdia															
57				mdhd														
58				hdlr														
59				minf														
60					gmhd													
61					hdlr													
62					dinf													
63						dref												
64					stbl													
65						stsd												
66							tmcd											
67						stts												
68						stsc												
69						stsz												
70						stco												
71	free																	
72	mdat																	

Figure 61. Comparison of Original GoPro Hero 3 and YouTube Encoded File Structure

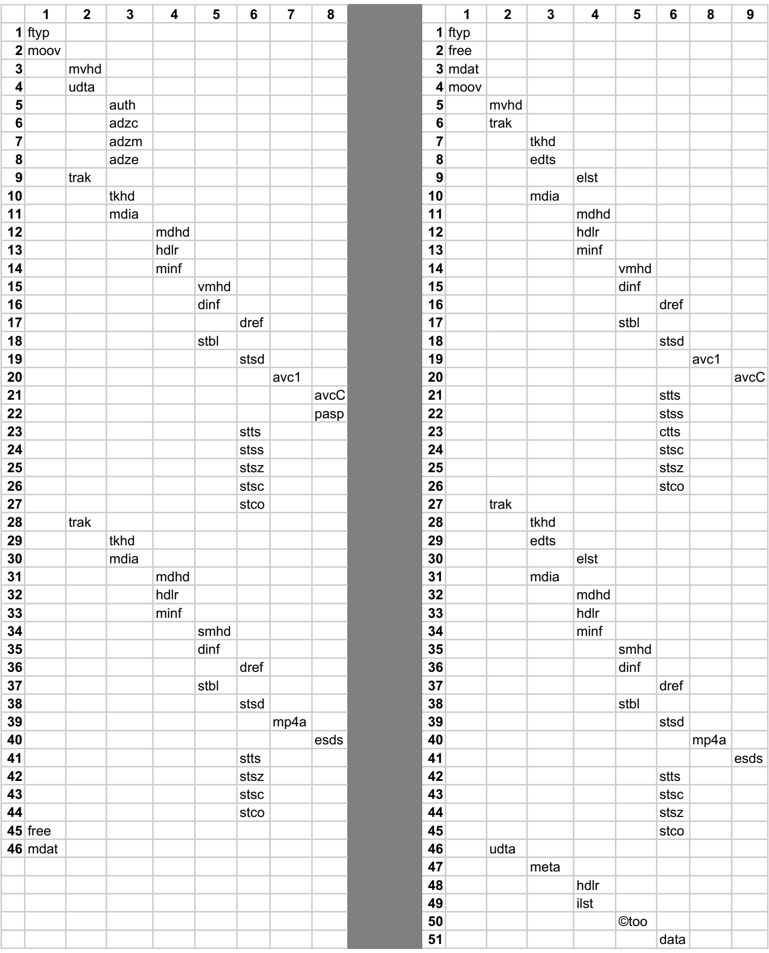
MediaInfo confirms relevant changes to the file properties of the re-encoded file. The format profile and codec have been modified and the embedded timestamps have been zeroed out. The presence of the 'Lavf56.25.101' string in this file correlates with the theory that youtube-dl is using ffmpeg to transcode YouTube's downloaded data stream into a playable format.



General		**General**	
Complete name	1920x1080-GOPRO-HERO3-GOPR1682-BL.mp4	Complete name	1920x1080_gopro_youtube.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	JVT	Format profile	Base Media
Codec ID	avc1	Codec ID	isom
File size	20.3 MiB	File size	4.82 MiB
Duration	6s 440ms	Duration	7s 245ms
Overall bit rate	26.5 Mbps	Overall bit rate	5 584 Kbps
Encoded date	UTC 2015-04-26 17:56:56	Encoded date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-04-26 17:56:56	Tagged date	UTC 1904-01-01 00:00:00
AMBA	□□□	Writing application	Lavf56.25.101
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	Main@L4.2	Format profile	High@L4.2
Format settings, CABAC	Yes	Format settings, CABAC	Yes
Format settings, ReFrames	1 frame	Format settings, ReFrames	3 frames
Format settings, GOP	M=1, N=8	Format settings, GOP	M=1, N=16
Codec ID	avc1	Codec ID	avc1
Codec ID/Info	Advanced Video Coding	Codec ID/Info	Advanced Video Coding
Duration	6s 440ms	Duration	7s 174ms
Bit rate mode	Constant	Bit rate	5 494 Kbps
Bit rate	25.0 Mbps	Width	1 920 pixels
Width	1 920 pixels	Height	1 080 pixels
Height	1 080 pixels	Display aspect ratio	16:09
Display aspect ratio	16:09	Frame rate mode	Variable
Frame rate mode	Constant	Frame rate	59.940 fps
Frame rate	59.940 fps	Minimum frame rate	59.920 fps
Color space	YUV	Maximum frame rate	59.960 fps
Chroma subsampling	4:02:00	Color space	YUV
Bit depth	8 bits	Chroma subsampling	4:02:00
Scan type	Progressive	Bit depth	8 bits
Bits/(Pixel*Frame)	0.201	Scan type	Progressive
Stream size	19.1 MiB (94%)	Bits/(Pixel*Frame)	1:03:22
Title	GoPro AVC	Stream size	4.70 MiB (97%)
Language	English	Encoded date	UTC 1904-01-01 00:00:00
Encoded date	UTC 2015-04-26 17:56:56	Tagged date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-04-26 17:56:56		
Color range	Full		
Color primaries	BT.709		
Transfer characteristics	BT.709		
Matrix coefficients	BT.709		
Audio		**Audio**	
ID	2	ID	2
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	6s 421ms	Duration	7s 245ms
Bit rate mode	Constant	Bit rate mode	Constant
Bit rate	128 Kbps	Bit rate	126 Kbps
Channel(s)	2 channels	Channel(s)	2 channels
Channel positions	Front: L R	Channel positions	Front: L R
Sampling rate	48.0 KHz	Sampling rate	44.1 KHz
Compression mode	Lossy	Compression mode	Lossy
Stream size	100 KiB (0%)	Stream size	111 KiB (2%)
Title	GoPro AAC	Encoded date	UTC 1904-01-01 00:00:00
Language	English	Tagged date	UTC 1904-01-01 00:00:00
Encoded date	UTC 2015-04-26 17:56:56		
Tagged date	UTC 2015-04-26 17:56:56		
Other			
ID	3		
Type	Time code		
Format	QuickTime TC		
Duration	6s 440ms		
Time code of first frame	17:55:51:27		
Time code, striped	Yes		
Language	English		
Encoded date	UTC 2015-04-26 17:56:56		
Tagged date	UTC 2015-04-26 17:56:56		

Figure 62. MediaInfo Comparison of Original GoPro Hero 3 and YouTube Encoded File

The original LG G3 video file uploaded to YouTube was also downloaded and analyzed. Its structure is consistent with the ffmpeg re-encoded videos examined for this paper and is distinctly different from the structure of an original LG G3 file.

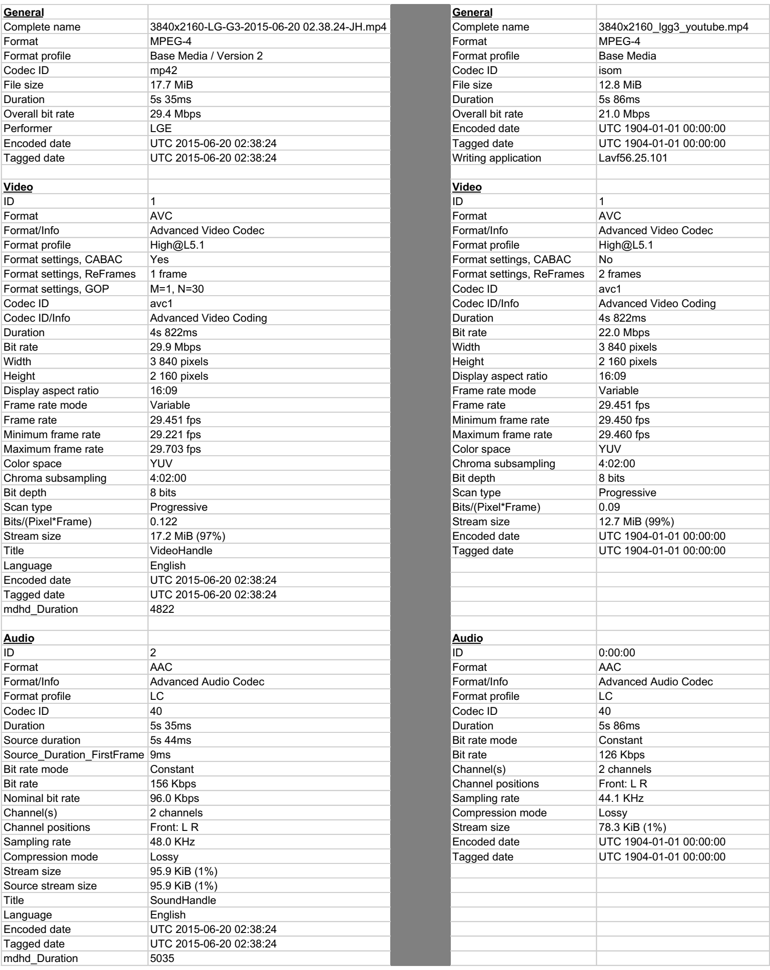


	1	2	3	4	5	6	7	8
1	ftyp							
2	moov							
3		mvhd						
4		udta						
5			auth					
6			adzc					
7			adzm					
8			adze					
9		trak						
10			tkhd					
11			mdia					
12				mdhd				
13				hdlr				
14				minf				
15					vmhd			
16					dinf			
17						dref		
18					stbl			
19						stsd		
20							avc1	
21								avcC
22								pasp
23						stts		
24						stss		
25						stsz		
26						stsc		
27						stco		
28		trak						
29			tkhd					
30			mdia					
31				mdhd				
32				hdlr				
33				minf				
34					smhd			
35					dinf			
36						dref		
37					stbl			
38						stsd		
39							mp4a	
40								esds
41						stts		
42						stsz		
43						stsc		
44						stco		
45	free							
46	mdat							

	1	2	3	4	5	6	8	9
1	ftyp							
2	free							
3	mdat							
4	moov							
5		mvhd						
6		trak						
7			tkhd					
8			edts					
9				elst				
10			mdia					
11				mdhd				
12				hdlr				
13				minf				
14					vmhd			
15					dinf			
16						dref		
17					stbl			
18						stsd		
19							avc1	
20								avcC
21						stts		
22						stss		
23						ctts		
24						stsc		
25						stsz		
26						stco		
27		trak						
28			tkhd					
29			edts					
30				elst				
31			mdia					
32				mdhd				
33				hdlr				
34				minf				
35					smhd			
36					dinf			
37						dref		
38					stbl			
39						stsd		
40							mp4a	
41								esds
42						stts		
43						stsc		
44						stsz		
45						stco		
46		udta						
47			meta					
48				hdlr				
49				ilst				
50					©too			
51						data		

Figure 63. Comparison of LG G3 Original and YouTube Encoded File Structure

As expected, MediaInfo reports the changes to format profile and codec ID, as well as the resetting of the embedded timestamps and presence of the ffmpeg identifying string in the metadata of the file.



General		**General**	
Complete name	3840x2160-LG-G3-2015-06-20 02.38.24-JH.mp4	Complete name	3840x2160_lgg3_youtube.mp4
Format	MPEG-4	Format	MPEG-4
Format profile	Base Media / Version 2	Format profile	Base Media
Codec ID	mp42	Codec ID	isom
File size	17.7 MiB	File size	12.8 MiB
Duration	5s 35ms	Duration	5s 86ms
Overall bit rate	29.4 Mbps	Overall bit rate	21.0 Mbps
Performer	LGE	Encoded date	UTC 1904-01-01 00:00:00
Encoded date	UTC 2015-06-20 02:38:24	Tagged date	UTC 1904-01-01 00:00:00
Tagged date	UTC 2015-06-20 02:38:24	Writing application	Lavf56.25.101
Video		**Video**	
ID	1	ID	1
Format	AVC	Format	AVC
Format/Info	Advanced Video Codec	Format/Info	Advanced Video Codec
Format profile	High@L5.1	Format profile	High@L5.1
Format settings, CABAC	Yes	Format settings, CABAC	No
Format settings, ReFrames	1 frame	Format settings, ReFrames	2 frames
Format settings, GOP	M=1, N=30	Codec ID	avc1
Codec ID	avc1	Codec ID/Info	Advanced Video Coding
Codec ID/Info	Advanced Video Coding	Duration	4s 822ms
Duration	4s 822ms	Bit rate	22.0 Mbps
Bit rate	29.9 Mbps	Width	3 840 pixels
Width	3 840 pixels	Height	2 160 pixels
Height	2 160 pixels	Display aspect ratio	16:09
Display aspect ratio	16:09	Frame rate mode	Variable
Frame rate mode	Variable	Frame rate	29.451 fps
Frame rate	29.451 fps	Minimum frame rate	29.450 fps
Minimum frame rate	29.221 fps	Maximum frame rate	29.460 fps
Maximum frame rate	29.703 fps	Color space	YUV
Color space	YUV	Chroma subsampling	4:02:00
Chroma subsampling	4:02:00	Bit depth	8 bits
Bit depth	8 bits	Scan type	Progressive
Scan type	Progressive	Bits/(Pixel*Frame)	0.09
Bits/(Pixel*Frame)	0.122	Stream size	12.7 MiB (99%)
Stream size	17.2 MiB (97%)	Encoded date	UTC 1904-01-01 00:00:00
Title	VideoHandle	Tagged date	UTC 1904-01-01 00:00:00
Language	English		
Encoded date	UTC 2015-06-20 02:38:24		
Tagged date	UTC 2015-06-20 02:38:24		
mdhd_Duration	4822		
Audio		**Audio**	
ID	2	ID	0:00:00
Format	AAC	Format	AAC
Format/Info	Advanced Audio Codec	Format/Info	Advanced Audio Codec
Format profile	LC	Format profile	LC
Codec ID	40	Codec ID	40
Duration	5s 35ms	Duration	5s 86ms
Source duration	5s 44ms	Bit rate mode	Constant
Source_Duration_FirstFrame	9ms	Bit rate	126 Kbps
Bit rate mode	Constant	Channel(s)	2 channels
Bit rate	156 Kbps	Channel positions	Front: L R
Nominal bit rate	96.0 Kbps	Sampling rate	44.1 KHz
Channel(s)	2 channels	Compression mode	Lossy
Channel positions	Front: L R	Stream size	78.3 KiB (1%)
Sampling rate	48.0 KHz	Encoded date	UTC 1904-01-01 00:00:00
Compression mode	Lossy	Tagged date	UTC 1904-01-01 00:00:00
Stream size	95.9 KiB (1%)		
Source stream size	95.9 KiB (1%)		
Title	SoundHandle		
Language	English		
Encoded date	UTC 2015-06-20 02:38:24		
Tagged date	UTC 2015-06-20 02:38:24		
mdhd_Duration	5035		

Figure 64. MediaInfo Comparison of LG G3 Original and YouTube Encoded File

CHAPTER VII

CONCLUSION

The framework for analysis outlined in this paper presents a viable means of authenticating a MPEG-4 recording based on its file structure and metadata. Test recordings from the device purported to have made the recording or a model of the same make and model will need to be created and analyzed in a forensically sound manner in order to establish the baseline of what constitutes an original file as created by the device. Once this baseline is established, that structure can be compared against the structure of the questioned file in order to determine authenticity.

In cases where the provenance of a questioned file is unknown, this framework of analysis presents a viable means of establishing a greater understanding of the file based on its file structure and metadata. If the file has been re-encoded due to editing, then the file's structure will be comparable to that of files created by known encoding software. To provide the greatest likelihood of identifying an unknown file, this framework of analysis could be utilized to create a database of file structures and properties from known devices and software encoders.

There are a number of open questions that present an opportunity for future work. Neither tool used in this method of analysis was created expressly for the purpose of forensic video analysis. It is important to explore the use of other existing tools for the purpose of analysis. Exiftool (http://www.sno.phy.queensu.ca/~phil/exiftool/) is a powerful tool for viewing image and video file. It supports MPEG-4 video containers and its use should be explored as an alternative or addition to MediaInfo. Another powerful tool that should be considered for further analysis is an extension of the ffmpeg project called ffprobe (https://ffmpeg.org/ffprobe.html) VLC (https://www.videolan.org/vlc/index.html) and GSpot (http://www.videohelp.com/software/GSpot) are two other tools that can report MPEG-4 file properties but It should be noted that none of these tools will report on the MPEG-4 container structure of a file, nor will they report on the contents of any forensically relevant containers of the file such as the User Data Box ('udta'). Defraser, a tool released by the Nederlands Forensisch Instituut (NFI), released under the BSD license at http://sourceforge.net/projects/defraser/, is a tool used to find video data streams in unallocated disk space. Its use to bolster this method of authentication should be explored as it is an actively maintained purpose-built tool for the purpose of forensic video examination.

In order to create a validated database of file structures from known devices, it will be important to create a new purpose built tool to parse the file structure of these files. This tool should also take into account and record the contents of the User Data Box ('udta'). None of the tools surveyed for this paper are capable of returning the contents of this forensically relevant container.

It is also important to expand the pool of video files to be analyzed. A larger collection of data will only serve to help refine the methods of analysis and reveal further similarities in file structure across device manufacturers. A study of the effects of software versions would also serve to help strengthen such a database. There are many open questions surrounding the idea of how device operating system software affects the file structure of recorded files. For example, does the file structure change across different versions of Android OS? An exploration of third party software would also help to identify if the file structure is created at the OS level of the device or by the software being used. The exploration of third party software would also allow the further analysis of the contents of the User Data Box (‘udta’) to determine what forensically relevant information recorded by a given piece of software.

As with any method proposed for the authentication of digital video, this method of authenticating digital video based on its file structure should be incorporated into a greater framework of digital video analysis that would correlate findings from as many analyses as possible in order to strengthen confidence in the ultimate opinion regarding a file’s authenticity. Digital video should be inherently more easily authenticated since there are two data streams to consider in analysis: the video and the audio. After the file structure and metadata have been analyzed for authenticity, further analysis can be performed on the pixel level of the video stream and at the sample level of the audio stream. By combining these three methods of analysis, I believe that a greater framework for digital video analysis can be realized.

REFERENCES

[1] Daniel Lawn Rappaport, “Establishing a Standard for Digital Audio Authenticity: A Critical Analysis of Tools, Methodologies, and Challenges.” University of Colorado Denver, 27-Apr-2012.

[2] Scott Dale Anderson, “Digital Image Analysis: Analytical Framework For Authenticating Digital Images.” University of Colorado Denver, 2011.

[3] T. Gloe, A. Fischer, and M. Kirchner, “Forensic analysis of video file formats,” *Proc. First Annu. DFRWS Eur.*, vol. 11, Supplement 1, no. 0, pp. S68–S76, May 2014.

[4] ISO/IEC, “ISO/IEC 11172-1:1993 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s -- Part 1: Systems.” ISO/IEC, 1993.

[5] ISO/IEC, “ISO/IEC 13818-1:1996 Information technology -- Generic coding of moving pictures and associated audio information -- Part 1: Systems.” ISO/IEC, 1996.

[6] ISO/IEC, “ISO/IEC 14496-1:1999 Information technology -- Coding of audio-visual objects -- Part 1: Systems.” ISO/IEC, 1999.

[7] ISO/IEC, “ISO/IEC 14496-2:1999 Information technology -- Coding of audio-visual objects -- Part 2: Visual.” ISO/IEC, 1999.

[8] ISO/IEC, “ISO/IEC 14496-3:1999 Information technology -- Coding of audio-visual objects -- Part 3: Audio.” ISO/IEC, 1999.

[9] Apple, Inc., “Classic Version of the QuickTime File Format Specification.” Apple, Inc., 2001.

[10] ISO/IEC, “ISO/IEC 14496-14:2003 Information technology -- Coding of audio-visual objects -- Part 14: MP4 file format.” ISO/IEC, 2003.

[11] ISO/IEC, “ISO/IEC 14496-10:2003 Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding.” ISO/IEC, 2003.

[12] ISO/IEC, “ISO/IEC 14496-15:2004 Information technology -- Coding of audio-visual objects -- Part 15: Carriage of network abstraction layer (NAL) unit structured video in ISO base media file format.” ISO/IEC.

[13] ISO/IEC, “ISO/IEC 14496-12:2004 Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format.” ISO/IEC, 2004.

[14] MP4 Registration Authority, “MP4REG Registered Types - File Types,” *MP4REG*, 15-Oct-2015. [Online]. Available: http://www.mp4ra.org/filetype.html. [Accessed: 15-Oct-2015].

[15] Apple, Inc., “QuickTime File Format Specification.” Apple, Inc., 2015.

[16] MP4 Registration Authority, “MP4REG Registered Types - Codecs,” *MP4REG*, 15-Oct-2015. [Online]. Available: http://www.mp4ra.org/codecs.html. [Accessed: 15-Oct-2015].

[17] MP4 Registration Authority, “MP4REG Registered Types - Box Types,” *MP4REG*, 15-Oct-2015. [Online]. Available: http://www.mp4ra.org/atoms.html. [Accessed: 15-Oct-2015].

[18] Gravity Lab, “What is the difference between Baseline, Main and High h264 mpeg4 / mp4 profiles?,” *GravityLab*. [Online]. Available: http://www.gravlab.com/2013/11/07/difference-baseline-main-high-h264-mpeg4-mp4-profiles/. [Accessed: 15-Oct-2015].

[19] Leach, et al., “A Universally Unique IDentifier (UUID) URN Namespace.” The Internet Society, Jul-2005.