Your Car is Recording: Metadata-driven Dashcam Analysis System

Kukheon Lee[a], Jong-Hyun Choi[a], Jungheum Park[a,*], Sangjin Lee[a]

[a]*Institute of Cyber Security and Privacy (ICSP), Korea University, 145 Anam-ro, Seongbuk-Gu, Seoul, South Korea*

Abstract

Dashcam as an on-board camera is useful as a source of potential digital evidence not only to reveal the truth of a traffic accident but also to explain the situation of a crime scene as a moving surveillance camera. It stores multimedia data as well as a variety of additional information needed for accident investigation including time, location, speed, accelerometer, etc. Under these circumstances, various studies have been conducted for dashcam forensics, but most of them focused mainly on extracting and interpreting visual video frames. In this paper, we identify and classify various metadata generated by 14 dashcam models produced by 11 manufacturers. Furthermore, we develop a normalized database schema to manage different multiple metadata and then discuss several dashcam forensic activities based on it. In addition, a prototype open-source tool is presented to support the proposed metadata-driven dashcam forensics.

Keywords: Dashcam, onBoardCam, Metadata, Container, Footage, Digital forensics

1. Introduction

Dashboard camera (Dashcam) is an embedded recording device that is mounted on a vehicle, in order to record driving and parking situations as well as store them as multimedia container files. It has become an essential element of vehicles according to changes of related laws and regulations. For example, in U.S., commercial vehicles such as police cars, buses and taxis install it to comply with *Commercial Motor Vehicle Safety*, and private vehicles including `Uber` and `Lyft` use it for reasons like preparing an evidence of accident investigation and resolving disputes on insurance claims [1, 2, 3]. Due to a considerable rise in the number of accidents and thefts, market analysts predict that the global dashcam market is expected to register a CAGR (Compound Annual Growth Rate) of 15.4% during the forecast period of 2019-2024, starting with USD 2,519 billion in 2018 [4].

In general, dashcam stores proprietary multimedia data in a removable storage device such as SD Card, and major manufacturers usually provide a dedicated viewer to enable examiners to analyze visual data along with non-visual information including time, location, speed, etc. Although several forensic tools provide advanced analysis functions especially for multimedia data, they are usually unable to thoroughly process a variety of dashcam models from multiple manufacturers. That is, though examiners rely on dedicated viewers from manufactures for analysis of additional information (metadata) pertaining to each video frame, they may have difficulty in responding to issues such as bankruptcy of manufacturers and corruption of files having proprietary formats. Under these circumstances, various studies have been conducted for dashcam forensics, but most of them focused mainly on extracting, recovering and analyzing visual frames. In other words, there is a lack of systematic research on non-visual metadata that can be stored in a standard or non-standard manner.

To respond this situation, we first identify and classify various metadata within 14 dashcam devices from the perspective of different levels of data management, including *filesystem* and *application*. This step includes reverse-engineering the storing structure as well as interpreting the meaning of each metadata. The systematized result of metadata as a high-level overview of dashcam forensics could be used as a reference in analyzing unknown dashcams. Next, we store the identified metadata in a normalized form, and then show that integrated metadata databases can support digital forensic activities. As examples of these activities, we discuss 'event reconstruction' based on a cross-metadata timeline generation, 'abnormal detection' through correlating multiple timestamps, and 'source identification' through extraction of unique characteristics from a set of metadata. Then, to demonstrate our proposal, we develop and open-source parsers and plugins for *plaso* [5] that can process dashcam related data including well-known multimedia formats, NMEA (National Marine Electronics Association) [6] records and customized proprietary data formats.

In summary, this paper makes the following contributions:

- Identifying metadata generated by 14 dashcams produced by 11 manufacturers
- Proposing a normalized database schema to store and manage multiple different metadata
- Introducing dashcam forensic activities using integrated metadata databases

*Corresponding author.
Email addresses: `kukheon1109@gmail.com` (Kukheon Lee), `antares0531@gmail.com` (Jong-Hyun Choi), `jungheumpark@korea.ac.kr` (Jungheum Park), `sangjin@korea.ac.kr` (Sangjin Lee)
URL: `http://forensic.korea.ac.kr` (Sangjin Lee)

- Demonstrating the proposed concept through developing an open-source tool based on *plaso*

The remainder of this paper is structured as follows: Section 2 summarizes backgrounds and existing studies relating to dashcam forensics. Section 3 describes target devices and experimental environments, and then Section 4 identifies and interprets various metadata generated by dashcams. Section 5 proposes a strategy for integrating and utilizing those metadata. Section 6 develops an open-source tool to prove our proposals. Finally, Section 7 present conclusions and future works.

2. Background and Related Works

2.1. Background

2.1.1. Basic operations of dashcam

Data storage. Dashcams mainly use flash memory based storage devices such as SD Card and Micro SD. Therefore, a storage device used in a dashcam has a relatively small capacity compared to one used in a host system like PC. In addition, it is not hard for ordinary users to access stored multimedia data, due to the characteristic as a *removable* storage device. For reference, it should be noted that there also exist *cloud-connected* dashcams that can upload multimedia files to linked online services, and it would be one of our future works.

File management. While a dashcam is operating, there may occur a situation where its storage device is full, due to 24/7 data generation. In this situation, dashcams usually delete the oldest multimedia file before creating a new one.

Recording mode. Dashcams support various recording modes, and each mode is used as a condition to begin recording at a specific time. Basically, there are two modes, *normal* (= driving) and *parking* based on whether a vehicle is started or not. There is also *event* mode for recording when a shock is detected from 3-axis accelerometer sensor. In addition, *manual* mode can be used to begin recording manually whenever users need it.

2.1.2. Structure of multimedia container files

A multimedia container format is used to store multiple objects (e.g., video, audio and text) in a single file. The MP4 (ISO Based Media File Format) [7] and AVI (Resource Interchange File Format) [8] are the most popular file formats to handle multimedia data. Both formats store multiple types of data using a specific unit such as *chunk* in AVI and *box* (or *atom*) in MP4. From here, we will use the term *chunk* to indicate a basic structure inside AVI and MP4 container files. In general, multiple chunks have a hierarchical structure for managing different types of data, and each chunk consists of *type*, *size* and *data* fields. It should be note that there are pre-defined standardized chunks, but developers can define their own chunks to store their own data. Due to this characteristic, different hardware or software may create various unknown chunks having unknown data structures.

2.1.3. Video codecs

Digital video devices use a codec to compress video data efficiently. The popular standards of codec for dashcmas are H.264 [9] and H.265 [10]. For efficient compression, a video data stream consists of a set of GOPs (group of pictures). As an example of H.264, there are several types of components as illustrated in **Figure 1**: *SPS* (Sequence Parameter Set) and *PPS* (Picture Parameter Set) that include decoding information, *IDR* (Instantaneous Decoding Refresh) that can be decoded independently, and *Non-IDR* that needs to reference adjacent frames to be decoded as a picture. These components manage data using NAL (Network Abstraction Layer) and RBSP (Raw Byte Sequence Payload) structures [11].

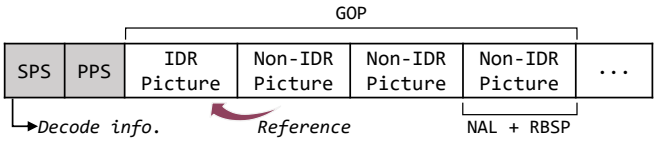

Figure 1: Simplified internal structure of an H.264 encoded data stream

2.2. Related Works

Although the previous studies on dashcams and multimedia files have been focused mainly on how to recover deleted or damaged video data [12, 13], there are several studies on analyzing multimedia container files for dashcam forensics.

Gloe et al. studied the structural characteristics of multimedia containers [14]. Their goal was source identification, distinguishing between original and post-processed videos, using the feature. They analyzed 19 digital camera models, 14 mobile phone models, and 6 video editing toolboxes. They used the following two features: (1) chunk hierarchy (sequence) in container files from each device, and (2) signature values that can be stored in special areas such as *INFO* chunk in AVI and non-standard chunks in MP4.

Song et al. proposed an algorithm for tampering detection using the internal structure of a video file [15]. They utilized chunk sequences generated by several video editing programs.

De Rosa et al. proposed a methodology for content-based multimedia forensics [16]. Their methodology consists of four steps: (1) metadata extraction and analysis, (2) audio-visual inspection, (3) results analysis, and (4) reporting. Although a multimedia forensic guideline was provided, it has limitations because it was based on only EXIF data.

An and Lee presented an analysis method for non-image data for 6 dashcam models [17]. They proposed the following 5-step procedure: (1) analysis on video data structures, (2) extraction and reconstruction of video streams, (3) analysis on non-visual data structures, (4) extraction and reconstruction of non-visual information, and (5) relevance analysis between visual (video) and non-visual information. This work has a limitation that they just proposed a concept without detailed explanation.

Cha et al. suggested requirements (operational functions) that a dashcam forensic tool should meet [18]. Through analyzing 10 dashcam models, they derived requirements for *analysis*, *management*, *video playback*, and *recovery* functions.

Table 1: List of dashcam devices for experiments

Manufacture	Model	Filesystem	Container	Origin	Release
Anycom	Thepoint	FAT32	AVI	KR	2015
BlackVue	DR4500L-FHD	FAT32	MP4	KR	2016
Dabonda	DBL-1000H	FAT32	MP4	KR	2014
Dabonda	DBH-4000H	FAT32	AVI	KR	2014
FineVu	CR-500HD	FAT32	AVI	KR	2013
iRoad	T-35	FAT32	AVI	KR	2013
iTronics	ITB-100SPW	FAT32	MP4	KR	2013
LUKAS	LK-7950WD	FAT32	AVI	KR	2015
Mercedes-benz	Starview	FAT32	AVI	-	2018
Thinkware	FXD900	FAT32	AVI	KR	2013
Thinkware	QXD3000	FAT32	MP4	KR	2018
Thinkware	Z500Plus	FAT32	MP4	KR	2019
TOUCHGO	K7	FAT32	AVI	KR	2018
Transcend	DrivePro220	FAT32/exFAT	MP4	TW	2015

Harjinder studied about the admissibility of the dashcam evidence submitted online and the metadata from dashcams [19]. This study summarizes the techniques to consider when submitting dashcam evidence online. Also, 7 dashcams were briefly analyzed to review the stored metadata. However, this study analyzed only results from several tools. As a result, the authors did not provide any details on internal structures of dashcam data.

As a result of literature review, we found that there has been no study in terms of thorough analysis of dashcam related metadata (i.e., non-visual information) to support practical forensic activities.

3. Experimental Environment

To select dashcam devices for experiments, this work considered several factors including *release date*, *supported filesystems*, *supported multimedia file formats*, *recording modes*, *number of camera channels*, and *watermarked text*. As a result, 14 dashcams from 11 manufacturers were selected, and **Table 1** shows summarized information on them.

After selecting target devices, each device was mounted on a vehicle to generate various events including 'driving' and 'parking', in order to create experimental datasets. During the process, we also considered more than two channels for some models. As shown in **Table 1**, 13 of 14 devices only support FAT32, but `Transcend`'s *DrivePro220* allows users to choose either FAT32 (default) or exFAT.

4. Dashcam Metadata Internals

A dashcam continuously generates and stores various types of data depending on conditions of a vehicle. This section identifies multiple metadata that can be found in dashcam's storage devices and describes their meaning for forensic activities.

4.1. Summary of findings

Figure 2 shows the proposed metadata-driven dashcam data analysis system. We identified 22 meaningful information (metadata) in various locations, through a systematic analysis of dashcam's storage devices from the perspective of different levels of data management, including *filesystem*, *operating system* and *application*. To represent the identified metadata, we define 7 main-levels (Level-1) and 21 sub-levels (Level-2) as shown in **Table 2**. In the following subsections, each metadata is represented using the defined abbreviation ($Lv1$ or $Lv1_{Lv2}$). For reference, **Appendix Table A2** provides an integrated view of the metadata found in this study.

Table 2: Taxonomy of dashcam metadata and their abbreviations

Level1	Level2	Description
Container (C)	cn	chunk name
	cs	chunk sequence
Device (D)	ch	channel
	mo	model
	ve	version
Location (L)	-	location
Recording mode (R)	ev	event (when shock)
	ma	manual
	no	normal (driving)
	pa	parking
Status (S)	gs	g-sensor
	sp	speed
	vo	voltage
Time (T)	da	date accessed
	dc	date created
	tc	time created (incl. date)
	tm	time modified (incl. date)
User preference (U)	fs	frame per second
	ln	language
	pl	profiles and levels
	re	resolution (width and height)
	tz	timezone

4.2. Filesystem-related metadata

As target models support well-known FAT32 and/or exFAT filesystems, it is possible to get useful metadata from *directory entries* of FAT.

4.2.1. Directory hierarchy

As shown in **Appendix Table A1**, the directory hierarchy where multimedia container files are stored allows examiners to identify specific recording modes (R) and timestamps (T) relating to the container files. Specifically, 12 models except *Starview* and *ITB-100SPW* use recording modes (R) for naming top-level directories, and 2 models (*DBH-4000H* and *T-35*) record creation dates (T_{dc}) of container files in sub-directory names. As each device generally creates a unique directory hierarchy, it is necessary to obtain device-specific information prior to dashcam forensics.

4.2.2. File naming rules

Under each of the directories identified above, there are basically multimedia container files, and each dashcam model uses a set of rules to assign names to the files. Therefore, through the name of a specific multimedia file, it is possible to get useful metadata relating to the file, including channel (D_{ch}), recording modes (R), time created (T_{tc}), etc.

Different models have different rules for file naming, as summarized in **Appendix Table A1**. Examiners can obtain R, T and

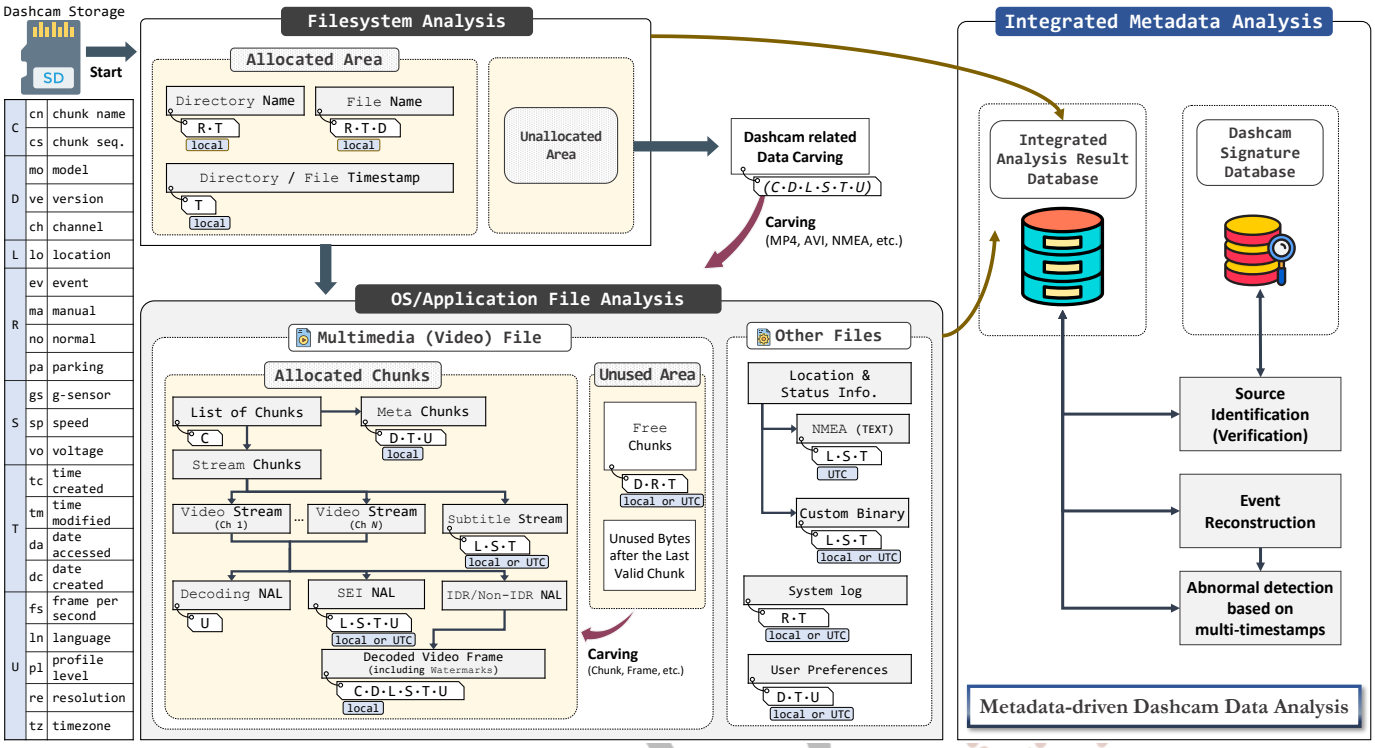

Figure 2: Metadata-driven dashcam data analysis (our findings and proposals)

D through file names generated by 13 devices except *DrivePro220*. For example, the following is the name assigned to a multimedia file from *DR4500L-FHD*:

2019-11-09-13-45-32_F_event.mp4

As shown in the name, 'YYYY-MM-DD-hh-mm-ss' means *creation time* (T_{tc}), which is usually the same as the creation timestamp value stored in an associated *directory entry* of FAT filesystem. The next 'F' stands for one of camera channels that are distinguished by using characters such as *S* (Single), *F* (Front), *R* (Rear) and *D* (Dual). The last 'event' indicates a certain condition for recording that can be one of *event* (when shock), *manual*, *normal* (driving) and *parking*. In summary, the name given as an example means that it has video data of the front camera when a shock was delivered at 2019/11/09 13:45:32 (Local).

4.2.3. Other considerations

In addition to directory/file names, 'timestamps' (including created, accessed and modified) stored in *directory entries* of FAT filesystem will of course be useful for forensic purposes. For instance, a creation timestamp can be used as a factor to determine whether a file was normally created by a dashcam or copied from an external system.

Moreover, due to the nature of 24/7 operation, multimedia files are frequently created and deleted in a storage device. So, examiners need to consider special recovery (carving) processes for residual data (i.e., deleted *directory entries* or operating system/application-related files) within unallocated areas of the storage device.

4.3. OS/Application-related metadata

Various types of metadata can also be found from operating system and application-related files.

4.3.1. Container files: list of chunks

Multimedia container formats generally store multiple types of data using a specific unit such as *chunk*. Thus, a multimedia container file may have unique chunk names (C_{cn}) as well as a chunk sequence (C_{cs}), according to the software to generate it [15]. That is, there is a possibility to identify a specific dashcam model or multimedia editing tool through those chunk related information. An example in **Figure 3** shows two lists of chunks generated by two different devices. As shown in the figure, container files from *T-35* and *LK-7950WD* have clearly different chunk hierarchies.

Our experiments revealed that all models supporting the MP4 format generate unique chunk sequences. However, in the case of several models (including *Thepoint*, *FXD900* and *K7*) that only support the AVI format, it is difficult to distinguish between individual models by using only a list of chunks, because they create an identical chunk sequence. The details of chunk names and sequences found in the remainder 12 dashcam models are shown in **Appendix Figure A1**.

4.3.2. Container files: meta chunks

There are special-purpose chunks that contain meaningful information from a digital forensic perspective. In particular, as depicted in **Figure 4**, it is important to note that *meta* chunks may include timestamps, resolution, number of frames, duration of video/audio data, device information, etc.

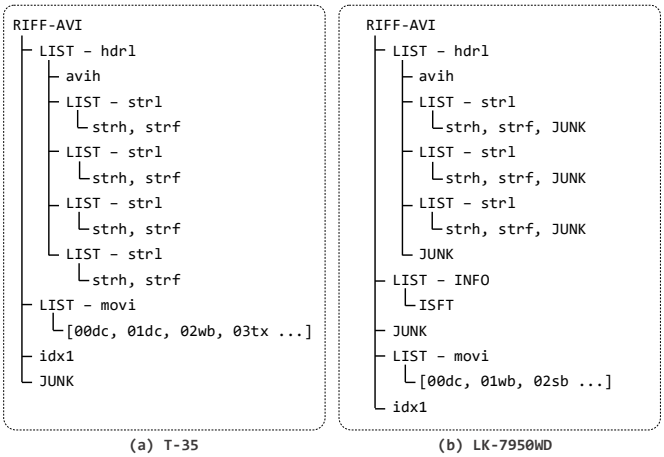

Figure 3: List of chunks created by two dashcam models

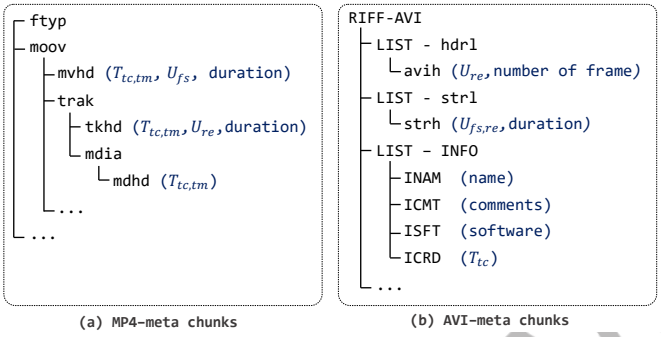

Figure 4: Meta chunks stored in MP4 and AVI container files

From MP4 containers, it is possible to identify various *meta* chunks including *mvhd* (movie header), *tkhd* (track header) and *mdhd* (media header), and they commonly contain created (T_{tc}) and modified (T_{tm}) timestamps in UNIX time format. The T_{tc} and T_{tm} mentioned here are usually the same, which allows examiners to understand when an MP4 container was first created. In addition, regarding AVI, there are some representative *meta* chunks including *avih* (avi header), *strh* (stream header) and sub-chunks of *INFO* list chunk (e.g., *ICRD*, *INAM* and *IPRD*). For example, an *INAM* chunk generated by *CR-500HD* contains 'CR-500HD' as its model name (D_{mo}) in ASCII string format.

For reference, it should be noted that the *INFO* list chunk may not exist in some AVI containers, because it is not mandatory in the format specification.

4.3.3. Container files: video stream

A video stream of a multimedia container file is an area in which a set of GOPs generated during a predetermined time period (usually 1 to 3 minutes) is stored. In other words, it is an area in which raw data containing actual visible information is saved, but some useful metadata can also be identified here.

The H.264 and H.265 codecs basically store information needed for decoding such as *frames per second* and *resolution* in SPS NAL [11]. Regarding AVI, raw data (including decoding information and a set of GOPs) generated by a codec is stored as it is in *##dc* (compressed video stream; *##* means a sequence number assigned to each stream) chunks under a *movi* list chunk. However, the MP4 format separately stores decoding information in an *avc1* (H.264) or *hvc1* (H.265) chunk within a *stsd* (sample description) chunk.

Interestingly, *CR-500HD* generates SEI (Supplemental Enhancement Information) NALs [9], and they are used to store metadata such as time (T), location (L) and status (S) in a proprietary unknown format defined by its manufacturer. For your guidance, the result of reverse-engineering its storing structure is as shown in **Figure 5**.

4.3.4. Container files: decoded video frame

It is possible to obtain video frames (a set of still images) by decoding a set of IDR and Non-IDR NALs within the above mentioned video stream. As a result of analysis on the decoded video frames, we confirmed that all 14 models insert information, including time (T), status (S), device (D) and user preference (U), into each frame as a watermark.

Figure 6 summarizes types of watermarks found in target models. Although positions vary, it can be seen that the composition of stored metadata is similar across different models. To effectively utilize this information for forensic activities, examiners can try to extract text from each frame by using OCR (Optical Character Recognition) engines such as *Tesseract* [11, 20].

4.3.5. Container files: subtitle stream

Most of the target models used in our experiments create a customized 'subtitle' stream to store metadata such as time (T), location (L), and status (S). Since it does not follow the standard subtitle formats, typical media players are unable to properly interpret the customized subtitle data.

For example, MP4 files created by several models including *DR4500L-FHD* have an independent track for *text*-type data to store additional information in their own format. In addition, in the case of AVI files created by several models such as *T-35*, there exists a *subtitle* chunk with names like *##tx*, *##sb* and *##st* within a *movi* list chunk, as shown in **Figure 7**. As mentioned above, *##* means a sequence number assigned to each stream. So, if there are two video streams (00 for *Front* and 01 for *Rear*) along with one audio stream (02), the lastly stored subtitle stream will be assigned as '03' as shown in the figure.

As a result of analysis, 11 of 14 models have a customized 'subtitle' stream, and 10 of them store metadata (L, S_{gs} and so on) in the standardized NMEA format. The other model, *DBL-1000H*, that does not contain a GPS module records S_{gs} in a customized binary format.

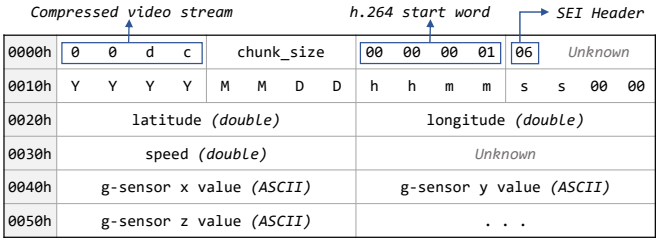

Figure 5: A proprietary SEI NAL data structure created by *CR-500HD*

model	[position: list of watermark]
Thepoint	[3:T_{tc},S_{gs},D_{ve}]
DR4500L-FHD	[3:D_{mo},D_{ve},T_{tc}],[4:S_{vo}]
DBL-1000H	[3:S_{gs},U_{fs}],[4:T_{tc}]
DBH-4000H	[3:D_{mo},T_{tc}]
CR-500HD	[3:S_{vo},S_{gs}],[4:U_{fs},S_{sp},T_{tc}]
T-35	[3:D_{ve}]
ITB-100SPW	[2:T_{tc}]
LK-7950WD	[3:T_{tc},S_{sp}],[4:S_{vo}]
Starview	[3:T_{tc},$S_{gs,vo}$,D_{ve},S_{sp}]
FXD900	[3:S_{vo}],[4:T_{tc}]
QXD3000	[3:$D_{mo,ve}$,S_{vo},T_{tc},S_{sp}]
Z500Plus	[3:$D_{mo,ve}$,S_{vo}, T_{tc},S_{sp}]
K7	[3:T_{tc},$S_{gs,sp,vo}$,D_{ve}]
DrivePro220	[3:D_{mo},T_{tc}]

Figure 6: Watermarks inserted in video frames

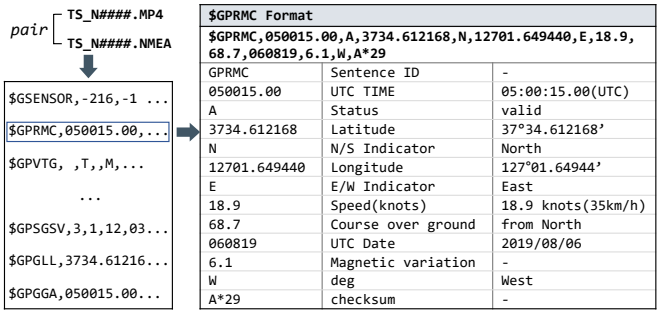

Figure 8: An example of .NMEA file created by *DrivePro220*

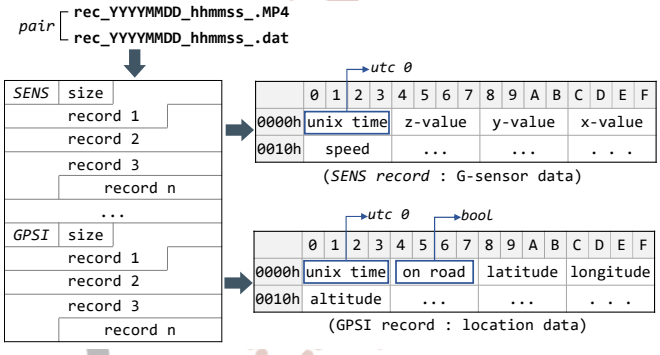

Figure 9: An example of .DAT file created by *ITB-100SPW*

4.3.6. *Container files: unused area*

Unused areas may exist inside a multimedia container file. Examples of the areas include the *JUNK* chunk of AVI we well as *free* and *skip* chunks of MP4, which can be used for padding. These chunks are also used for special purposes, and there have been cases where device information (D) and creation timestamp (T_{tc}) are stored [14]. Among the target devices tested here, an AVI file produced by *T-35* or *Starview* has an interesting characteristic of storing its file name in a *JUNK* chunk.

Additionally, container files with a 'fixed' length may have unused areas outside their valid ranges [11]. For example, an MP4 file from *QXD3000* utilizes the lastly located *free* chunk to manage the remainder of the valid data.

While these unused areas are not related to the playback of multimedia files, they are likely to include parts of meaningful data (e.g., meta chunks, video streams, subtitle streams, etc.) that has been generated in the past. Therefore, it needs to be confirmed by applying 'carving' techniques specially designed for dashcam related data.

4.3.7. *Other files*

Location and status information. Dashcam devices also generate other files than multimedia container files. Here, we introduce (1) a text file (`.NMEA`) where NMEA information is stored and (2) a custom data file (`.DAT`) specially defined by a manufacture. First, a `.NMEA` file is created to record additional information about video frames for each multimedia container file. As shown in **Figure 8**, it allows examiners to identify location (L), time (T), and status (S) through interpreting text-based standard NMEA messages. Next, customized data structures from manufactures are also used rather than standardized formats, like indicated in **Figure 9**. Our reverse-engineering results revealed that `.DAT` files in *ITB-100SPW* manage a set of location (L) and status ($S_{gs,sp}$) values by using two proprietary data structures, *SENS* and *GPSI*.

System logs. Some models including *T-35* record detailed operational events in a text-based log file. As an example, **Figure 10** shows a part excerpted from a system log, having multiple event entries such as *power on/off*, *storage device usage*, *file creation* and so on.

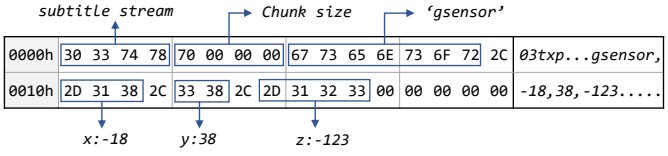

Figure 7: An example of g-sensor data structure in an AVI container file

User preferences. Additional metadata can also be identified from files where user preferences (configurations) are stored. As a result of analyzing the target models, we confirmed that dashcams generally use standard formats like SQLite and XML for managing device's HW/SW information, video quality, audio volume, screen brightness, language, etc. As an example, **Figure 11** indicates an XML-based file from *T-35*.

5. Integration and Analysis of Dashcam Metadata

This section defines a database schema for integrating dashcam metadata introduced in the previous section, and discusses its usefulness through three examples of digital forensic activities: *event reconstruction*, *abnormal detection* and *source identification*.

5.1. *Integration*

To efficiently store and manage dashcam metadata, we define a database schema as illustrated in **Figure 12**.

In this schema, 'device_info' is a table for managing basic information (e.g., manufacture, model and version) on dashcam devices. Tables used for directories ('directory_info') and files ('file_info') in a storage device is composed of fields for information from both normal and recovered entries. In addition, metadata ('metadata') and time ('time_value') tables have *data_type* and *data_source* fields, in order to manage various metadata introduced in this paper.

```
[2019/11/09 09:49:46] Power On
[2019/11/09 09:49:46] Insert MMC/SD card
[2019/11/09 09:49:56] Recording Start
[2019/11/09 09:50:06] Create /mnt/mmc/NORMAL/20191109_093930/20191109_095006_I2.tmp
                                    ...
[2019/11/09 09:53:07] Closed /mnt/mmc/NORMAL/20191109_093930/20191109_095207_I2.avi
[2019/11/09 09:59:53] ------------------------------------------------------------
[2019/11/09 09:59:53] Enter Parking Mode
[2019/11/09 09:59:54] Closed /mnt/mmc/NORMAL/20191109_093930/20191109_095909_I2.avi
                                    ...
[2019/11/09 15:24:27] Exit Parking Mode
[2019/11/09 15:24:27] ------------------------------------------------------------
[2019/11/09 15:24:37] Create /mnt/mmc/NORMAL/20191109_125200/20191109_152437_I2.tmp
[2019/11/09 15:24:42] Closed /mnt/mmc/NORMAL/20191109_125200/20191109_152437_I2.avi
[2019/11/09 15:24:43] Recording Stop
[2019/11/09 15:25:26] Recording Start
```

Figure 10: An example of system logs managed by *T-35*

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright(c) DABONDA. CO., LTD. All right reserved. -->
<setup>
 <!-- Product Information -->
 <item name="productName" value="IROAD"/>
 <item name="modelName" value="T35"/>
 <item name="swVersion" value="1.0.5"/>
 <item name="hwVersion" value="0.0"/>
 <!-- Settings -->
 <item name="videoQuality" value="2"/> <!-- 0: Normal, 1: High, 2: Highest -->
 <item name="audioInEnable" value="false"/>
                                    ...
  <!-- Language Settings -->
    <item name="language" value="KOR"/> <!-- KOR, ENG, RUS, CHN, ESP, JPN -->
    <!-- Default Settings -->
    <item name="factoryReset" value="false"/>
</setup>
```

Figure 11: An example of XML-based configurations used in *T-35*

5.2. Analysis: event reconstruction

From a digital forensics perspective, it is possible to reconstruct dashcam related events in detail through a set of metadata that includes 'timestamps'.

Based on the proposed normalized database, we can create a timeline by extracting information about when (T), why (R), where (L) and how (S) events occurred. As listed in **Figure 13**, creating a metadata-driven timeline not only enhances traditional analysis activities focused on visible data, but also makes examiners effectively reconstruct events that occurred at the scene of an incident.

5.3. Analysis: abnormal detection based on multi-timestamps

In Section 4, this paper explained that multiple time values can be identified from multiple locations in a storage device. That is, there exist different time values corresponding to a variety of actions such as *occurrence of an event*, *creation/deletion of a container file*, *creation of a video frame*, etc. As a result of analysis, since these values are associated with each other, it is likely that the distribution of timestamps for a multimedia file is not normal, if the file was not perfectly manipulated. The following lists time values relevant to a specific multimedia container:

- *fn* (file name) has a T_{tc} value
- Parent directory's *dn* (directory name) has a T_{dc} value
- *de* (directory entry in FAT) has T_{tc} and T_{tm} values
- *mc* (meta chunks) has a set of T_{tc} and T_{tm} values
- Video frames' *wt* (watermark) has a set of T_{tc} values
- Video frames' *gi* (GPS information) has a set of T_{tc} values
- *sl* (system log) has a T_{tc} value

Among them, *wt* and *gi* include continuous timestamps from the time a file was first created until the time it was last saved. Especially, *gi* can be extracted from various sources including SEI NALs, subtitle stream and other files (.NMEA and .DAT). The relationship between time values mentioned above is represented as follows:

$$dn \leq fn = de_{tc} = sl \leq mc_{tc,tm} \leq wt = gi <= de_{tm}$$

The above conditions can be used to validate distributions of time values identified from a dashcam device. It should be noted that only timestamps (*mc*, *wt*, *gi*) inside multimedia container formats need to be considered for individual multimedia files recovered from unallocated areas of a storage device.

5.4. Analysis: source identification

In multimedia forensics, 'source identification' is one of research subjects that are being actively studied in terms of *reliability of evidence*. On the basis of unique characteristics of metadata introduced in the previous section, the following features can be considered for source identification: a) folder hierarchy and file naming rules, b) chunk sequence, and c) watermark.

As a result of analysis, there are limitations in accurate identification of source dashcam devices, if each feature is considered alone. First, it is difficult to distinguish multiple models from a manufacturer, only through 'folder hierarchy' and 'file naming rules'. For example, as listed in **Appendix Table A1**, *Thinkware*'s models use the same naming rule to create container files. Second, there are several dashcam models that are difficult to distinguish even when 'chunk sequence' is used. As an instance, *K7*, *Thepoint* and *FXD900* have the same chunk sequence as shown in **Appendix Figure A1**. Third, it is also hard to use 'watermark' only to classify exactly all dashcam models. Of course, it will be possible if a watermark text includes a specific model name, but only 5 of 14 models have such information within each decoded video frame.

Therefore, it is necessary to utilize combinations of multiple metadata for practically distinguishing dashcam models. In our preliminary experiment, a simple combination of chunk sequence, watermark and decoding information (i.e., resolution and compression rate) was enough to identify sources of container files created by all 14 models. To expand it, our future studies will focus on deriving an efficient metadata combination for source identification through analyzing additional dashcam devices, and will also implement the proposed concept of the *dashcam signature database* as shown in **Figure 2**.

6. Implementation

6.1. Design concept

Based on the above mentioned results, to support the proposed metadata-driven dashcam forensics, we design an automated tool to process dashcam related data. The tool is divided into two parts as depicted in **Figure 14**: *pre-processing* and *analysis*. The first part is the step of parsing dashcam-related

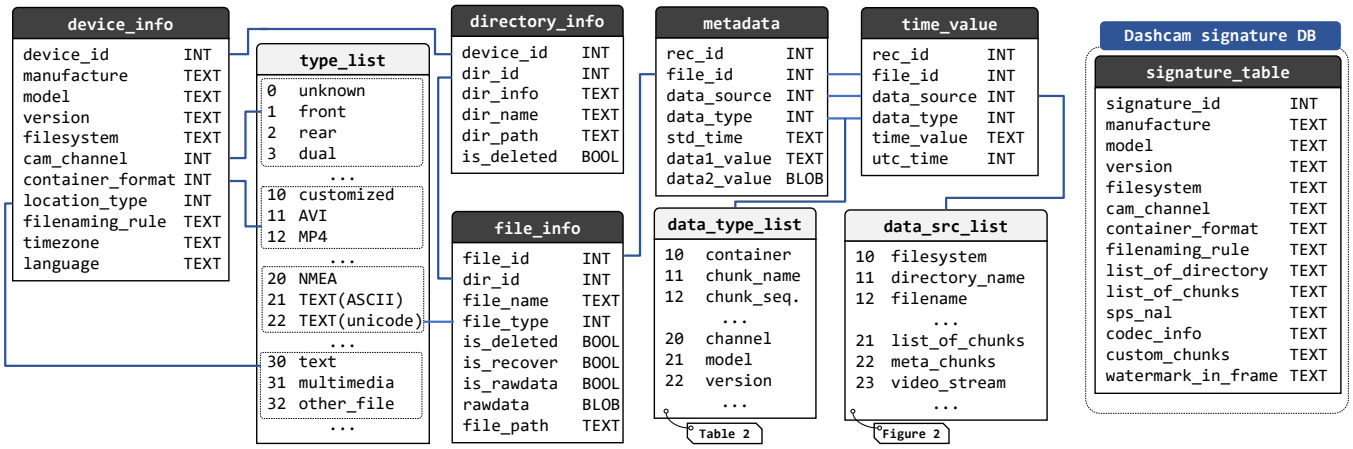

Figure 12: A proposed database schema designed to support metadata-driven dashcam data analysis

No	Time	Timezone	source	type	data
1	2020/01/05 14:20:30	UTC	filesystem	time created	-
2	2020/01/05 14:20:30	UTC	meta_chunks	time created	-
3	2020/01/05 14:20:30	UTC	filename	record.mode	normal
4	2020/01/05 14:20:31	UTC	subtitle	location	37.583230, 127.027490
5	2020/01/05 14:20:31	UTC	subtitle	status	0.5,-0.4,-0.6
6	2020/01/05 14:20:31	UTC	subtitle	speed	14
n			...		
n+1	2020/01/05 14:25:10	UTC	filename	record.mode	event
n+2	2020/01/05 14:25:10	UTC	subtitle	location	37.583099, 127.027152

Figure 13: An example of event reconstruction for a dashcam

files and extracting metadata by using *plaso*. Because the current version of *plaso* does not have the ability to handle multimedia data, developing new parsers and plugins is required to support this part. Specifically, it should be able to process standard multimedia formats (i.e., AVI and MP4), NMEA records and customized proprietary data formats, as introduced in Section 4. Next, the second part is to perform filtering, interpreting and integrating dashcam-related records from an output of *plaso*. As a result of these analysis operations, a normalized result database is created, as well as an existing dashcam signature database is updated if necessary.

6.2. Prototype implementation

We are implementing the design concept described above as an integrated automated tool. To demonstrate our proposal, the current version of a Python-based prototype is released as an open source project[1].

7. Conclusion and future work

This paper identified various types of metadata within 14 dashcam models from the perspective of different levels of data management. This work also proposed a normalized database schema to store and manage multiple different metadata, and introduced possible forensic activities using integrated dashcam metadata databases. Then, based on these findings, this paper developed an open-source tool to demonstrate the proposed metadata-driven dashcam forensic analysis.

Since the current version of *plaso* does not have the ability to process multimedia files, we expect the results of this study to be useful in the digital forensic community. Consequentially, the proposed metadata-driven approach will not only enhance traditional visual data-oriented analysis but can also be utilized as a foundation for dashcam forensics in new perspectives.

There are several plans for enhancing the proposed concept, such as investigating additional dashcam models from diverse manufacturers, identifying further metadata from them, deriving an efficient metadata combination for source identification, and developing a stable tool for supporting efficient dashcam forensics.

[1] [GitHub URL] https://github.com/kukheon1109/Dashcam_Plaso

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2018-0-01000, Development of Digital Forensic Integration Platform)

References

[1] UBER, Can I use a video camera (2019 (accessed January 05, 2020)). URL https://help.uber.com/partners/article/can-i-use-a-video-camera-?nodeId=efaad152-cbb6-45fe-9d7d-911842d21c8b

[2] Lyft, Recording device policy (2019 (accessed November 20, 2019)). URL https://help.lyft.com/hc/en-us/articles/115012923127-Safety-policies

[3] D. Štitilis, M. Laurinaitis, Legal regulation of the use of dashboard cameras: Aspects of privacy protection, Computer Law & Security Review 32 (2) (2016) 316–326.

[4] M. Intelligence, Dashboard camera market - growth, trends, and forecast (2019 - 2024), Tech. rep., Mordor Intelligence (2019).

[5] K. Gudjonsson, plaso ((accessed March 05, 2020)). URL https://github.com/log2timeline/plaso

[6] N. M. E. Association, et al., NMEA 0183–Standard for interfacing marine electronic devices, NMEA, 2002.

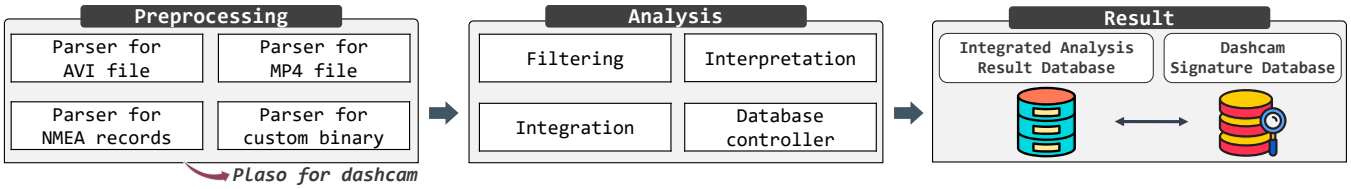

Figure 14: A design concept for implementation

[7] ISO, Iso/iec 14496-12, information technology — coding of audio-visual objects — part 12: Iso base media file format (2020).
[8] Microsoft, AVI RIFF File Reference (2018 (accessed May 3, 2018)). URL `https://docs.microsoft.com/en-us/windows/win32/directshow/avi-riff-file-reference`
[9] T. ITU, Advanced video coding for generic audiovisual services, ITU-T Recommendation H. 264.
[10] T. ITU, H.265: Infrastructure of audiovisual services – coding of moving video – high efficiency video coding, ITU-T Recommendation H. 265.
[11] J. Park, S. Lee, Data fragment forensics for embedded dvr systems, Digital Investigation 11 (3) (2014) 187–200.
[12] K. Alghafli, C. Y. Yeun, E. Damiani, Techniques for measuring the probability of adjacency between carved video fragments: The vidcarve approach, IEEE Transactions on Sustainable Computing.
[13] G.-H. Na, K.-S. Shim, K.-W. Moon, S. G. Kong, E.-S. Kim, J. Lee, Frame-based recovery of corrupted video files using video codec specifications, IEEE Transactions on Image Processing 23 (2) (2013) 517–526.
[14] T. Gloe, A. Fischer, M. Kirchner, Forensic analysis of video file formats, Digital Investigation 11 (2014) S68–S76.
[15] J. Song, K. Lee, W. Y. Lee, H. Lee, Integrity verification of the ordered data structures in manipulated video content, Digital Investigation 18 (2016) 1–7.
[16] T. Gloe, Forensic analysis of ordered data structures on the example of jpeg files, in: 2012 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2012, pp. 139–144.
[17] H. An, S. Lee, The analysis of data structure to digital forensic of dashboard camera, Journal of the Korea Institute of Information Security and Cryptology 25 (6) (2015) 1495–1502.
[18] I. H. Cha, K. H. Lee, S. J. Lee, Design and implementation of car black-box forensic analysis tool through the analysis of data structure, KIPS Transactions on Computer and Communication Systems 5 (11) (2016) 427–438.
[19] H. S. Lallie, Dashcam forensics: A preliminary analysis of 7 dashcam devices, Forensic Science International: Digital Investigation 33 (2020) 200910.
[20] R. Smith, An overview of the tesseract ocr engine, in: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Vol. 2, IEEE, 2007, pp. 629–633.

Appendix

This appendix contains extra tables and figures for reporting detailed findings.

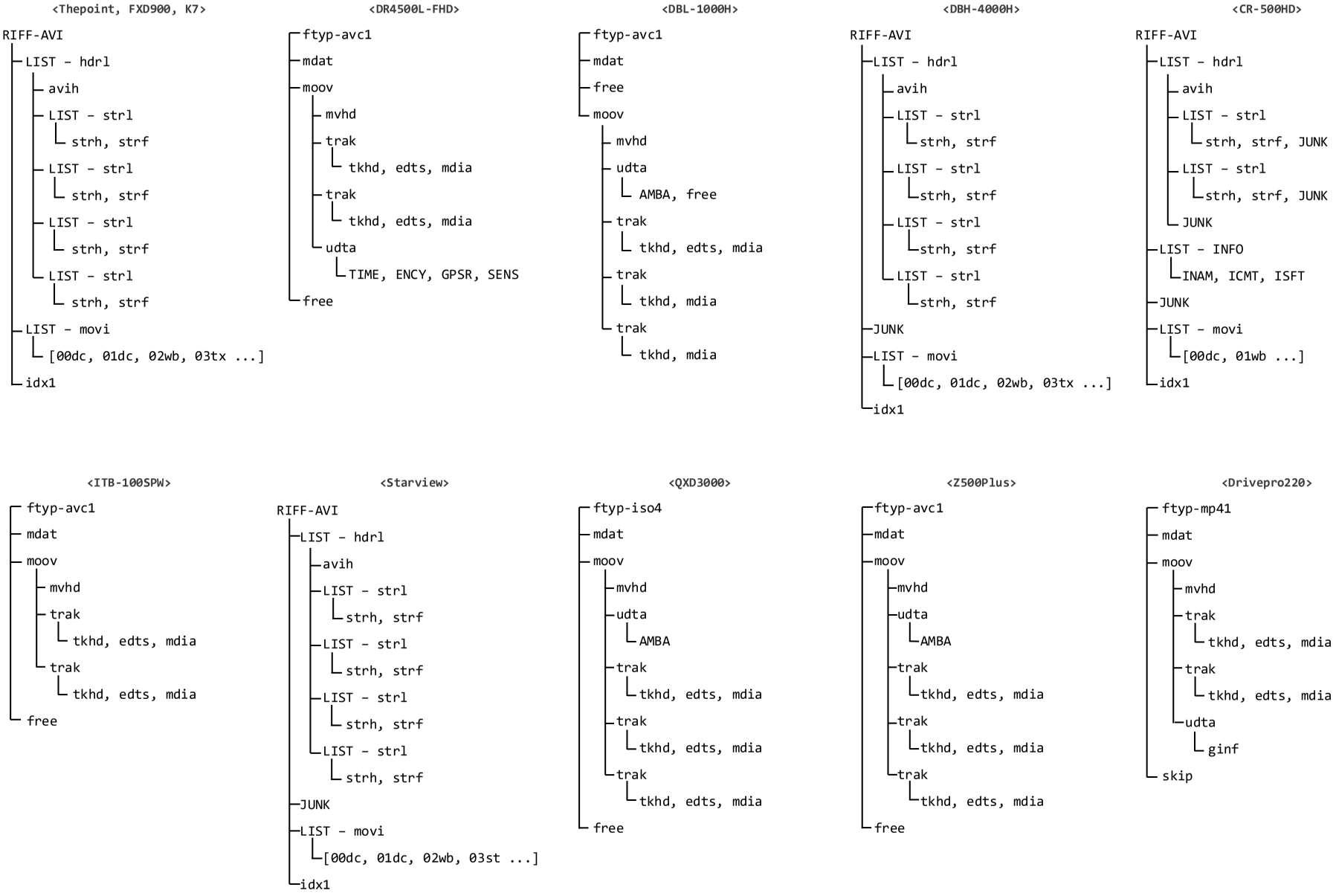

Figure A1: List of chunks created by 12 dashcam models

Table A1: Directory hierarchy (including directory naming rules) in storage devices used by dashcams

Model	Category	Description
Anycom Thepoint	Dir.	PARKING(*pa*), SETTING, USER(*ma*), DRIVING(*no*), EVENT(*ev*)
	File	RM_YYYYMMDD_hhmmss_CH.avi (e.g., Rec_20191109_134532_D.avi)
	Note	RM: Park(*pa*), Manual(*ma*), Rec(*no*) or Event(*ev*), CH: of S(single) or D(dual)
BlackVue DR4500L-FHD	Dir.	BlackBox(*no*), Event(*ev*), Parameter, Parking(*pa*), ParkingEvent(*ev*)
	File	YYYY-MM-DD-hh-mm-ss_CH_RM.mp4 (e.g., 2019-11-09-13-45-32_F_normal.mp4)
	Note	CH: F(front) or R(rear), RM: normal(*no*), event(*ev*), motion(*pa*) or parking(*ev*)
Dabonda DBL-1000H	Dir.	SETUP, EVENT(*ev*), NORMAL(*no*)
	File	YYYYMMDD_hhmmss_RM.mp4 (e.g., 20191109_134532_INF.mp4)
	Note	RM: EVT(*ev*) or INF(*no*)
Dabonda DBH-4000H	Dir.	config, event(*ev*), normal(*no*)
	Sub-dir.	YYYYMMDD(*dc*)
	File	nnnnn_YYYYMMDD_hhmmss_RM.avi (e.g., 00003_20191109_134532_N.avi)
	Note	nnnnn: sequence number, RM: E(*ev*) or N(*no*)
FineVu CR-500HD	Dir.	camcorder(*ma*), event(*ev*), normal(*no*), photo(*ma*)
	File	YYYY-MM-DD-hh'h'mm'm'ss's'_RM.avi (e.g., 2019-11-09-13h54m32s_normal.avi)
	Note	RM: manual(*ma*), parking(*pa* in event dir.), normal_to_event(*ev*), event(*ev*), motion(*pa* in event dir.), normal(*no*) or capture(*ma*)
iRoad T-35	Directory	EVENT(*ev*), NORMAL(*no*), PARK(*pa*), LOG
	Sub-dir.	YYYYMMDD(*dc*)
	File	YYYYMMDD_hhmmss_RMCH.avi (e.g., 20191109-134532_I2.avi)
	Note	RM: E(*ev*), I(*no*) or M(*pa*), CH: 1(Single) or 2(Dual)
iTronics ITB-100SPW	Directory	Setup
	File	RM_YYYYMMDD_hhmmss_.mp4, RM_YYYYMMDD_hhmmss_.dat (e.g., rec_20191109_134532_.mp4)
	Note	RM: rec(*no*), evt(*ev*) or pmr(*pa*) [Note that .dat file is for storing location and status info.]
LUKAS LK-7950WD	Directory	AlwaysMovie(*no*), MotionMovie(*pa*), EventMovie(*ev*), obd
	File	RM_YYYYMMDD_hhmmss_CH.avi (e.g., alwa_20191109_134532_F.avi)
	Note	RM: alwa(*no*), motion(*pa*) or event(*ev*), CH: F(front) or R(rear)
Mercedes-benz Starview	Directory	root directory only
	File	RM_YYYYMMDD_hhmmss_CH.avi (e.g., REC_20191109_134532_F.avi)
	Note	RM: REC(*no*), EVT(*ev*), MAN(*ma*) or PAR(*pa*), CH: F(front) or R(rear)
Thinkware FXD900	Directory	driveinfo, DRIVEX, Driving(*no*), Driving Shock(*ev*), Manual(*ma*), Parking(*pa*), Parking Shock(*ev*), Safety Box, SETTING
	File	RM_YYYY_MM_DD_hh_mm_ss_CH.avi (e.g., REC_2019_11_09_13_45_32_D.avi)
	Note	RM: REC(*no*), EVT(*ev*), MAN(*ma*), PMOT(*pa*) or PEVT(*ev* in Parking Shock dir.), CH: S(single) or D(dual)
Thinkware QXD3000	Directory	driveinfo, DRIVEX, Driving(*no*), Driving Shock(*ev*), Manual(*ma*), Parking(*pa*), Parking Shock(*ev*), Safety Box, SETTING
	File	RM_YYYY_MM_DD_hh_mm_ss_CH.mp4 (e.g., REC_2019_11_09_13_45_32_F.mp4)
	Note	RM: REC(*no*), EVT(*ev*), MAN(*ma*), MOT(*pa*) or PAK(*ev* in Parking Shock dir.), CH: F(Front) or R(Rear)
Thinkware Z500Plus	Directory	driveinfo, DRIVEX, Driving(*no*), Driving Shock(*ev*), Manual(*ma*), Parking(*pa*), Parking Shock(*ev*), Safety Box, SETTING
	File	mode_YYYY_MM_DD_hh_mm_ss_CH.mp4 (e.g., REC_2019_11_09_13_45_32_F.mp4)
	Note	RM: REC(*no*), EVT(*ev*), MAN(*ma*), MOT(*pa*) or PAK(*ev* in Parking Shock dir.), CH: F(Front) or R(Rear)
TOUCHGO k7	Directory	CFIG, CONT(*no*), EVNT(*ev*), PARK(*pa*), PEVT(*ev*), USER(*ma*)
	File	RM_YYYYMMDD_hhmmss_CH.avi (e.g., Cont_20191109_134532_D.avi)
	Note	RM: Evnt(*ev*), Cont(*no*), User(*ma*) or Park(*pa*), CH: S(single) or D(dual)
Transcend DrivePro220	Directory	DCIM(*ma*), N-VIDEO(*no*), P-VIDEO(*ev*), SYSTEM
	File	TS_RMnnnn.mp4, TS_RMnnnn.NMEA (e.g., TS_N0003.mp4)
	Note	RM: D(*ma*), N(*no*) or P(*ev*), *nnnn*: sequence number [Note that .NMEA file is for storing location and status info.]

※ This table uses abbreviations for Level-2 metadata listed in **Table 2**.
※ RM: recording mode, CH: channel, YYYY: year, MM: month, DD: day, hh: hour, mm: minute, ss: second.

Table A2: Summary of dashcam metadata

Model	Filesystem: directory name	Filesystem: file naming rule	Filesystem: unallocated area	Container: meta chunk	Container: video stream	Container: decoded video frame[2]
Thepoint	R	$D_{ch} \cdot R \cdot Ttc$	ALL[1]	$U_{fs,re}$	$U_{fs,pl,re}$	$D_{ve} \cdot T_{tc} \cdot S_{gs}$
DR4500L-FHD	$R_{ev,no,pa}$	$D_{ch} \cdot R_{ev,no,pa} \cdot T_{tc}$	ALL	$T_{tc,tm} \cdot U_{fs,pl,re}$	-	$D_{mo,ve} \cdot T_{tc} \cdot S_{vo}$
DBL-1000H	$R_{ev,no}$	$R_{ev,no} \cdot T_{tc}$	ALL	$T_{tc,tm} \cdot U_{fs,pl,re}$	-	$S_{gs} \cdot U_{fs} \cdot T_{tc}$
DBH-4000H	$R_{ev,no} \cdot T_{dc}$	$R_{ev,no} \cdot T_{tc}$	ALL	$U_{fs,re}$	$U_{fs,pl,re}$	$D_{mo} \cdot T_{tc}$
CR-500HD	$R_{ev,ma,no}$	$D_{ch} \cdot R \cdot T_{tc}$	ALL	$D_{mo,ve} \cdot U_{fs,re}$	$L \cdot S_{gs,sp} \cdot T_{tc} \cdot U_{fs,pl,re}$	$S_{vo,gs} \cdot U_{fs} \cdot S_{sp} \cdot T_{tc}$
T-35	$R_{ev,no,pa} \cdot T_{dc}$	$D_{ch} \cdot R_{ev,no,pa} \cdot T_{tc}$	ALL	$U_{fs,re}$	$U_{fs,pl,re}$	D_{ve}
ITB-100SPW	-	$R_{ev,no,pa} \cdot T_{tc}$	ALL	$T_{tc,tm} \cdot U_{fs,pl,re}$	-	T_{tc}
LK-7950WD	$R_{ev,no,pa}$	$D_{ch} \cdot R_{ev,no,pa} \cdot T_{tc}$	ALL	$U_{fs,re}$	$U_{fs,pl,re}$	$T_{tc} \cdot S_{sp,vo}$
Starview	-	$D_{ch} \cdot R \cdot T_{tc}$	ALL	$U_{fs,re}$	$U_{fs,pl,re}$	$T_{tc}, S_{gs,vo} \cdot D_{ve} \cdot S_{sp}$
FXD900	R	$D_{ch} \cdot R \cdot T_{tc}$	ALL	$U_{fs,re}$	$U_{fs,pl,re}$	$S_{vo} \cdot T_{tc}$
QXD3000	R	$D_{ch} \cdot R \cdot T_{tc}$	ALL	$T_{tc,tm} \cdot U_{fs,pl,re}$	-	$D_{mo,ve} \cdot S_{vo} \cdot T_{tc} \cdot S_{sp}$
Z500Plus	R	$D_{ch} \cdot R \cdot T_{tc}$	ALL	$T_{tc,tm} \cdot U_{fs,pl,re}$	-	$D_{mo,ve} \cdot S_{vo} \cdot T_{tc} \cdot S_{sp}$
K7	R	$D_{ch} \cdot R \cdot T_{tc}$	ALL	$U_{fs,re}$	$U_{fs,pl,re}$	$T_{tc} \cdot S \cdot D_{ve}$
DrivePro220	$R_{ev,ma,no}$	$R_{ev,ma,no}$	ALL	$T_{tc,tm} \cdot U_{fs,pl,re}$	-	$D_{mo} \cdot T_{tc}$

Model	Container: subtitle stream	Container: unused area (AVI-junk)	Container: unused area (MP4-free,skip)	Other files: location and status info.	Other files: system log	Other files: user preference[3]
Thepoint	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-	-	-	$D_{mo,ve}$
DR4500L-FHD	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	ALL	-	-	D_{mo}
DBL-1000H	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-	-	-	$D_{mo,ve} \cdot T_{tm}$
DBH-4000H	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-	-	-	$D_{mo,ve}$
CR-500HD	-	-	-	-	-	$D_{mo,ve}$
T-35	$L \cdot S_{gs,sp} \cdot T_{tc}$	$D_{ch} \cdot R_{ev,no,pa} \cdot T_{dc}$	-	-	$R \cdot T_{tc}$	D_{mo}
ITB-100SPW	-	-	ALL	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	$D_{mo,ve}$
LK-7950WD	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-	-	$R \cdot T_{tc}$	$D_{mo,ve}$
Starview	$L \cdot S_{gs,sp} \cdot T_{tc}$	$D_{ch} \cdot R \cdot T_{tc}$	-	-	-	-
FXD900	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-	-	-	$D_{mo,ve} \cdot U_{tz}$
QXD3000	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	ALL	-	-	$D_{mo,ve} \cdot U_{tz}$
Z500Plus	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	ALL	-	-	$D_{mo,ve} \cdot U_{tz}$
K7	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-	-	$R \cdot T_{tc}$	$D_{mo,ve} \cdot U_{tz}$
DrivePro220	-	-	ALL	$L \cdot S_{gs,sp} \cdot T_{tc}$	-	-

※ In this table, only *Lv*1 means that there exist all Level-2 metadata (i.e., $R = R_{ev,ma,no,pa}$).

※ This table does not include *list of chunks* (C) column because all models commonly have the information.

[1] 'ALL' means that any types of metadata can potentially exist in unallocated or unused areas.

[2] Each list of metadata is presented by watermarked patterns shown in **Figure 6**.

[3] *Starview* and *Drivepro* do not store user preference related files on external storage devices. We assume that the devices manage preference (configuration) data using internal flash memory (e.g., NAND, eMMC) rather than external SD card.