Forensic Science International: Digital Investigation 44 (2023) 301500

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 EU - Selected papers of the Tenth Annual DFRWS Europe Conference

Analysis of real-time operating systems' file systems: Built-in cameras from vehicles

Jung-Hwan Lee, Bum-Su Hyeon, Oc-Yeub Jeon, Nam In Park*

Digital Analysis Division, National Forensic Services, 10 Ipchoon-ro, Wonju-si, Gangwon, 26460, Republic of Korea

ARTICLE INFO

Article history:

Keywords:
Vehicle forensic
Smart car
Built-in camera
File system
Reverse engineering
Real-time operating system

ABSTRACT

The vehicles we use daily record enormous information via the embedded system that is installed. This recorded information identifies traffic accidents and might reveal the truth of crimes committed using the vehicle. However, as the embedded systems mounted on vehicles must maintain real-time performance, they often have a real-time operating system (RTOS)-dedicated file system structure. Analysis of such systems is becoming a new challenge for digital forensics. This study analyzed the real-time operating system-dedicated file system of a vehicle's built-in camera. When the built-in camera of the vehicle operates normally, there are various ways to acquire video data. However, when the built-in cam circuit is damaged, the only available method is to extract the onboard memory. To analyze the dedicated file system, we analyzed the driver file in the system area using the reverse engineering technique. We could analyze various log files and user-setting files in multiple partitions stored in the analyzed memory. In addition, we proved that more video frames can be restored by extracting the unallocated area of the video storage partition. In the future, this method can be applied to analyze various RTOS and dedicated file systems installed in the vehicle.

1. Introduction

Vehicles are changing from mechanical and physics-centric devices in the past to electronic and software-centric devices these days. Newer vehicles are installed with dashcam systems, navigation devices, driving assistance systems, and vehicle safety systems. Even autonomous driving and telematics/infotainment systems are being released. As these systems value real-time and reliability in the connection of sensors and various electric devices, the network inside the vehicle mainly uses controller area network (CAN) communication, a dedicated network protocol. Each system in the past used simple embedded systems to maintain low latency (Daily et al., 2015).

However, in recent years, many real-time operating systems (RTOSs) have been used to perform scalability and more diverse functions (Seo et al., 2017). In such mobile environments, RTOS-embedded devices use the onboard type flash memory as the main storage media as they have to respond to physical shocks and various temperature environments, simultaneously guaranteeing a long life span.

Therefore, file systems in vehicles use a dedicated file system considering high responsiveness and flash memory characteristics.

To increase the compatibility/safety between the onboard type flash memory and the RTOS, the use of a dedicated file system is increasing (Zhang et al., 2020). Therefore, to perform digital forensic work on a vehicle that has been transformed into a digital device, it is necessary to target a different RTOS environment for each manufacturer and model.

Therefore, a digital forensic approach to embedded devices in this special environment (operating system, file system, storage medium, and information structure) is a new challenge (Kopencova and Rak, 2020). As it provides indisputable evidence in the analysis of crimes using vehicles, vehicle accidents, and accidents related to autonomous driving, further research in this field is necessary.

In this paper, we introduce approaches to data analysis and restoration, targeting the built-in video recording devices of HYUNDAI/KIA/GENESIS. In addition, a reverse engineering method is implemented to analyze the file system of the onboard type flash memory in the RTOS environment.

Consequently, video files obtained by the chip-off method in the damaged built-in video recording devices can be extracted using reverse engineering. Finally, we compared the video file extracted

* Corresponding author.
E-mail address: namin.park@gmail.com (N.I. Park).

https://doi.org/10.1016/j.fsidi.2023.301500

with the file system driver provided by the manufacturer and the video file analyzed by our reverse engineering technique. Moreover, we show it is possible to extract more video frames through the extraction of the unallocated area and to analyze user behavior files.

In a previous study various operating systems/file systems were introduced as an extensive review of the vehicle's audio video navigation (AVN) and event data recorder (EDR) systems (Le-Khac et al., 2020). These various file systems are generally unanalyzable by the existing digital forensic tools. Furthermore, in another study various dashcam file systems, video file structures, and metadata analysis were conducted, but the built-in video recording devices in the RTOS environment were not included (Lee et al., 2021a). Another study proposes a file system to improve the performance of flash memory storage media suitable for video recording environments (Kim and Shin, 2015). A flash-aware journaling technique was proposed and a study was conducted to improve the flash memory performance in video recording devices. Another study describes the difference between the current vehicle forensic procedure and the existing digital forensics. In addition, it suggested that the adoption of RTOSs in vehicle ECUs is expected to increase (Buquerin et al., 2021).

The present study introduces an approach for many electronic control units (ECUs).

2. Related work

2.1. Dash cam

Many ECU devices are installed in the vehicle, but among them, the dash cam provides obvious evidence in traffic accident analysis. The dash cam is installed inside and outside the vehicle and can store multiple channels of video; therefore, it is an essential analysis target for vehicle forensics (Kopencova and Rak, 2020). The dash cam records videos in various situations (e.g., parking, normal driving, and event driving). Moreover, by storing not only video but also audio, it can be used as major evidence to clearly reveal the cause of traffic accidents and criminal acts involving the vehicle. In addition, the dash cam is evolving into a device that can store location, time, and speed through GPS connection, and a safety function is implemented by applying image processing technology to maintain the vehicle in the same lane and to measure the distance from the vehicle in front.

Until recently, such dash cameras have been manufactured and installed by a third-party company rather than a vehicle manufacturer. Moreover, the dash cam mainly uses an embedded Linux operating system, stored in a read-only memory (ROM) drive. Hence, the vehicle stores video data file in external memory (Lee et al., 2021a). The reason of the choices is user convenience and system scalability. Most, of the external memories use the FAT series or customized file systems. For the FAT-formatted memory, the video files stored in the external memory can be separated and connected to an ordinary PC for access to the recorded data in the form of a video file. In the case of a customized file system, after parsing the file system through a dedicated viewer, the data can be read by reprocessing it into a common video file format. However, the vehicle market is changing from the external dash cam produced by the third party to the internal dash cam produced by the traditional vehicle company because camera modules are installed in various driving assistance devices (Around View, Lane Keeping Assist System, Adaptive Cruise Control and Auto Park) (Buquerin et al., 2021). For example, in the first case of TESLA vehicles, vehicles recognize the situation around the vehicle by recording three channels on the front, four channels on the side, and one channel on the rear. In vehicles released after 2021, an internal video channel is also added to monitor the driver's situation. These video data are used for vehicle control (Full-Self Driving). Among the eight-channel video data recorded in the FSD process, TESLA provides four-channel video data to be recorded according to user selection. Eventually, drivers will be able to use the dash cam built in the vehicle (TESLA, 2022).

In the second case of HYUNDAI/KIA/GENESIS vehicles, the vehicles launched after 2020 are also equipped with built-in cams. Such a built-in cam stores the video of both the front and rear channels, and does not record audio data (HYUNDAI, 2022). Fig. 1 shows the simplified components (G-sensor, AP chip, eMMC memory) of the HYUNDAI/KIA/GENESIS built-in cam. This device is called the drive video recording system (DVRS) by the manufacturer. This DVRS's eMMC memory stores the video data. The video data can be exported by inserting a USB thumb drive or connecting to the DVRS's access point (AP). These dash cams are mounted inside the vehicle in a built-in form and have characteristics different from the dash cams manufactured by third-party companies that have been used in the past.

The characteristics of the external dash cam created by the third-party companies and the built-in dash cam created by the vehicle companies are different, as shown in Table 1. The onboard memory of the built-in cam configured in this way is different from the existing external dash cam digital forensic method in the steps of identifying - data acquisition - file system parsing - data analysis - reporting.

2.2. RTOS and the dedicated file system

The built-in dash cam is difficult to analyze in the system analysis stage because it uses an operating system different from that in the existing desktops, laptops, and mobiles. Most early vehicle systems operated in the form of a monolith of the software installed in the ECU. Systems that have been used in vehicle ECUs since 2000 include Kenwood, Windows Embedded, VxWorks, and QNX (Buquerin et al., 2021).

Recently, as the number of ECUs for vehicles has increased and their roles have been diversified, RTOSs are being used to maintain real-time while ensuring the scalability and security of the OS and hypervisors (Cotroneo et al., 2021). In general, RTOSs are designed with priority-based scheduling, by separating partitions according to their roles using a microkernel system. This design concept is widely used in vehicle, satellite, and aviation systems because of the advantage that systems in other partitions can operate normally even when there is a problem in a specific partition. RTOSs used in vehicles are commercially available products, and systems with reliability and safety features are currently in use (Seo et al., 2017).

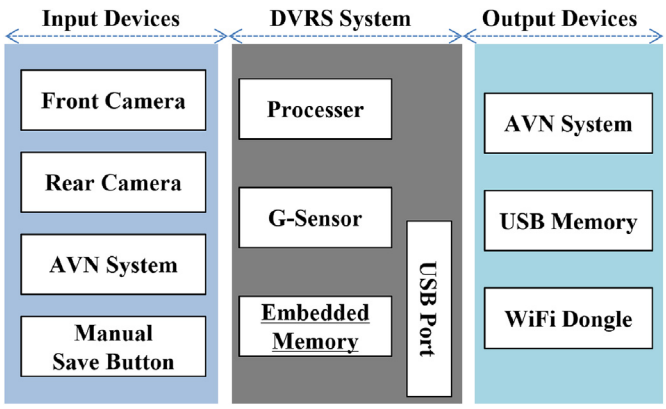

Fig. 1. Overview of the DVRS architecture.

Table 1
External dash cam Vs. Build in dash cam.

	External dash cam	Build in dash cam
Memory type	microSD/SD card	Onboard eMMC/USB thumb drive
Real-time guarantee	None	Must guarantee
Externally connected device	GPS antenna	AVN, GPS antenna, various sensors, ECU, etc.
Reliability information	Low (independent operation)	High (dependent operation
Life span	Up to memory life	Up to vehicle life

These systems should be able to support information flow control, resource isolation, trusted initialization, trusted delivery, trusted recovery, and audit capabilities. Various ECUs installed in vehicles use various RTOSs according to manufacturers and models (Buquerin et al., 2021). Most RTOSs use their own file systems to ensure reliability, increasing the overall diversity of the relevant file systems. Moreover, recently, manufacturers are designing their own operating systems.

- The Mercedes-Benz Group aims to release the Mercedes-Benz operating system (MB.OS) in 2024 (Mercedes-Benz, 2022).
- The BMW Group intends to install BMW Operating System 8, an extension of Android Automotive OS (AAOS) beginning in 2023 (BMW, 2022).
- VW Group is going to launch vw.os and Volkswagen Automotive Cloud system by 2025 (Volkswagen, 2022).
- Hyundai Group has developed ccOS using Nvidia's RIVE platform and aims to launch vehicles quipped with ccOS beginning in 2022 (Hyundai, 2022).

It is expected that more new file systems will be appear as the OSs designed by each vehicle manufacturer are diverse. Therefore, the existing digital forensic tools retain inherent limitations in their analysis coverage.

This paper presents methods to analyze the dedicated file system used in the built-in cam (DVRS module). It is necessary to understand the file system analysis method as it is expected that the systems targeted for vehicle forensics will be diversified in the future.

2.3. File system for flash memory

The file system used in the vehicle dash cam must be prepared for sudden-power-off (SPO) in the case of an accident. The file system most frequently used in dash cam by third-party manufacturers is the file allocation table (FAT)-based file system.

In the FAT-based file system, the metadata includes a file allocation table and directory entries, and a storage medium allocation area is managed in cluster units. The FAT is responsible for managing the array of entries for each allocated area of the cluster. Directory entries store the name, file size, first content cluster number, date/time information, etc., and are modified when created, modified, or accessed (Lee et al., 2021a). However, if SPO occurs at an important moment, such as in a car accident, or when the vehicle battery is discharged below the minimum operating power of the dash cam device, video and audio data storage is affected.

To solve this problem, many file systems for DVRS have appeared. Among them, the file systems targeting flash memory applied a page unit storage method and wear leveling techniques to evenly write cells, creating a file system configuration without a directory entry area that managed the FAT clusters. Similar techniques are applied to the dedicated file system that is analyzed in this study.

In addition, in DVRS systems that repeatedly store high-capacity data such as video data, file fragmentation often occurs when the FAT file system is used. If file fragmentation is severe, writing or reading a file slows down dramatically and memory is used inefficiently. This phenomenon becomes a serious problem in DVRS, which continuously stores and deletes high-capacity data. To prevent this phenomenon, the technique of applying the journaling structure to the existing file system(HFAT), the method of recording video data only in the partitioned area by dividing all the storage capacity in advance (ACPA), and a pre-division method using flash memory blocks of the same size as cluster groups (FECA), are suggested (Lee et al., 2021a). In particular, the structure of the on-board flash memory used by the built-in dash cam is difficult to replace.

Therefore, the built-in cam's file system prepares for SPO, does not cause file fragmentation, and requires a dedicated file system that provides real-time performance. Owing to these characteristics, many forensic tools that analyze the existing file system cannot read the partitioned file system.

3. Experimental environment (DVRS)

The HYUNDAI/KIA/GENESIS launched with built-in DVRS installed on and over 34 such models are released since 2019 (HYUNDAI, 2022). The DVRS can record the videos of the front/rear camera channels installed in the vehicle in real-time, and as shown in Fig. 1, it includes a function to automatically save the videos to the event folder in the event of a vehicle impact, using the G-sensor. In addition, it is linked with the AVN system to enable the user control over DVRS, and video playback and management. The built-in camera system has front/rear cameras and CAN communication ports for both input and output devices, respectively, a screen output terminal connected to the AVN device, and a USB terminal for downloading videos, as shown in Fig. 1.

The storage medium of DVRS is equipped with an onboard embedded multimedia card (eMMC) memory, as shown in Fig. 1. The name of the model of this memory manufactured by Samsung Electronic is “KLMCG8GESD-B04Q″ and the detailed specifications have been confirmed, as listed in Table 2.

3.1. Data acquisition methods

There are three methods for extracting data from the eMMC memory card.

Table 2
DVRS's eMMC specification (Samsung, 2022).

Type	Value
Version	eMMC5.1
Voltage	1.8/3.3 V
Density	64 GB
Interface	HS400
MLC	2-bit MLC
Temperature	−40–105 °C
Package size	11.5 × 13 × 1.0 mm (FBGA153)

- When the vehicle is turned on normally, the image data can be extracted via the USB port by selecting the desired video on the screen of the AVN system, as shown in the normal method in Fig. 2.
- If the vehicle is damaged and the power cannot be normally connected to the DVRS module or cannot be connected to the AVN system, data can be acquired through the "emergency download" method provided by the manufacturer, as shown in the emergency download method in Fig. 2.
- The eMMC memory chip can be desoldered when the DVRS board itself is damaged and the internal circuit is not normal even when the power is connected, as shown in the chip-off method in Fig. 2.

This method must physically separate the memory chip from the board to which it is soldered. But it is.

The only available method to acquire data if the normal and emergency download methods cannot be used. In the "normal mode" of extracting files using the USB drive in Fig. 2, you can selectively extract files parsed from the file system. However, it is impossible to extract file logs and unallocated spaces. In the case of extracting data in the "emergency download mode" of Fig. 2, connect power to the specific power port and connect the USB drive directly to the circuit board USB port. In this process, all files in the file system are copied to the connected USB drive, and the status can be checked by writing the log file as a text file on the USB drive. Of course, it is impossible to extract the file log and unallocated spaces in this process. The contents and results of the "emergency download method" are described in detail in 'A Case Study on Using Car AVN System' (Lee et al., 2021b).

3.2. Acquisition of data by the chip-off method

When the DVRS board is damaged and the emergency download protocol does not operate, the onboard eMMC memory must be desoldered from the board. Vehicle DVRS is more difficult to chip-off than other circuit boards due to the coating providing some protection against water and dust. The reason for this is that conventional embedded devices (Drone, IOT and GPS devices) are often uncoated because their expected service life is shorter than DVRS. Therefore, in order to remove the eMMC memory, the coating of the board must be removed before proceeding.

Then, we blew about 300 °C of wind up and down for 180 s to remove the chip. Then, some of the glue and lead were cleaned on the pin of the chip. During this process, it should take care not to let too much heat into the memory. Finally, the pin of the chip was revolved using liquid lead to improve connectivity.

We acquire the binary data of the entire area of the chip off memory using a socket suitable for the memory type and referring to the datasheet of the memory.

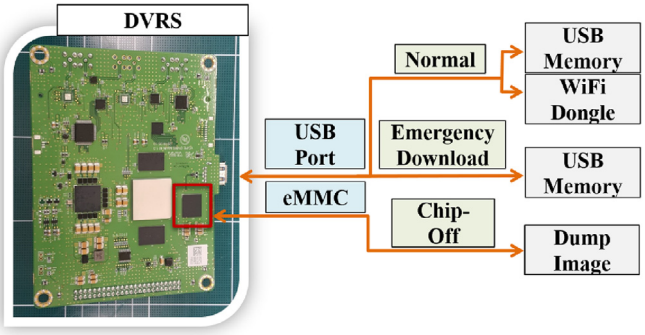

Fig. 2. Method of extraction of files from DVRS.

The data obtained in this way contains 56 partitions. Each partition performs the designated roles (backup, recovery, cache, system, user data, boot, video, update, etc.), benefitting isolation of each component of the RTOS. Among the partitions, there are partitions formatted in FAT and EXT file systems, as shown in Table 3, but most of the memory space is not a part of a known file system that can be interpreted by digital forensic tools.

4. Proposed methods for DVRS file system analysis

Most of the obtained partitions were not analyzed, and video file carving/frame-by-frame restoration was attempted for the acquisition of image data (Na et al., 2013).

- As a result of video file carving, most files are fragmented and cannot be restored.
- As a result of restoring frame by frame using known sequence parameter set (SPS)/picture parameter set (PPS), the image data was recorded discontinuously because of file fragments.

In the frame unit restoration method, time information cannot be recoded; therefore, it is difficult to apply the recombination method using time info.

4.1. Manufacturer-supplied file system package

Most of the partitions identified in Table 3 could not be analyzed with the existing digital forensic tools, and the ASCII byte sequence "TFFS" is recorded in beginning of each partition, as shown in Fig. 3.

By analyzing the acquired image with the string" TFFS", the information related to Tuxera Flash File System was found among the system files. We load the Tuxera Flash File System kernel module on Ubuntu 20.04 LTS (Kernel 5.4) x86_64 platform environment by receiving the package for parsing the file system from the manufacturer. The memory was mounted as Tuxera Flash File System "mount -t tffs" This enabled partition parsing. If the partition is not read normally, the partition volume can be checked and restored through "tffsck" provided by the manufacturer (Tuxera, 2022). However, related functions are not provided for the extraction of unallocated spaces and the restoration of deleted data required by digital forensics.

As a result of mounting the largest partition (yvideo) and checking the files, folders "databases", "systemfile", "thumb", and are acquired form the partition in addition to the folder obtained in the emergency download mode.

Table 3
List of major file systems in DVRS.

Name	File system	Size(KB)
bluetooth	FAT16	1024
boot	Android boot image	65,536
cache	Unknown(TFFS)	262,144
dsp	EXT4	16,384
modem	FAT16	97,280
recovery	Android boot image	65,536
recovery	Android boot image	65,536
persist	Unknown(TFFS)	32,768
userdata	Unknown(TFFS)	786,432
yapp	Unknown(TFFS)	307,200
ybackup	Unknown(TFFS)	786,432
ydata	Unknown(TFFS)	307,200
yupdate	Unknown(TFFS)	786,432
yvideo	Unknown(TFFS)	28,672,000

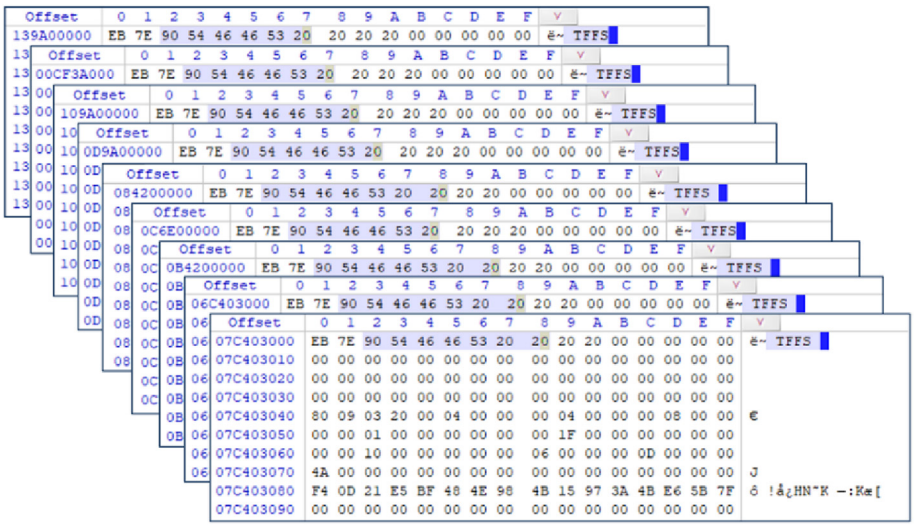

Fig. 3. "TFFS" signature written to each partition.

- In the "database" folder, a DB file is recorded using the SQLite format. It contains information about the files created when the DVRS G-sensors detects an impact.
- In the "systemfile" folder, setting files related to the DVRS device are recorded, and the file contains the DVRS control information changed according to the user settings.
- In the "thumb" folder, there are JPEG files in the "EVT", "MAN", "NOR", and "PAK" folders, and the number of these JPEG files is the same as the number of video files with the same name.

4.2. Our reverse engineering method

The dedicated file system of which structures and operational information is not known can be analyzed through reverse engineering. Our reverse engineering process is divided into four steps.

- Identifying the file system
- Acquiring of file system management software
- Analyzing of the operation of the software
- Specifying a file-system structure specification (Section 5)

4.2.1. Identifying the file system

File systems often write a magic number int the header part of the volume, normally the boot sector, to indicate the file system type. This magic number denotes the name or acronym of the file system in ASCII code. Therefore, if we find a string at the beginning of the target volume, it is most likely that the string names that file system. We obtain additional information, such as the characteristics of the file system, by searching for this string as a keyword in the whole disk images.

In general, if a keyword exists in/etc/fstab file where the mount configurations are stored in Unix-like operating system, the keyword can be confirmed as the name of the file system. In addition, information in the file system (e.g., manufacturer and version information) can be obtained from copyright statements and software information strings in the disk image.

4.2.2. Acquiring the file system management software

The software that manages the file system includes kernel drivers, formatting tools, error checking, and correction programs. As each software performs different processes, their cross-analysis provides richer information. In the case of a Unix-like operating system, this software is all packaged in ELF files, they can be acquired through keyword search and carving method.

4.2.3. Analyzing of the operation of the software

The acquired file system-related software is analyzed through a disassembler such as IDA PRO and Ghidra, and a debugger such as GDB (Govin et al., 2015).

Here, we assume that we want to find how the creation time of a file can be extracted. We first aim to analyze the code of the file system driver. Unix-like file system drivers are packaged in ELF files with the extension ".ko". we find the code related to the *sync* operation in the ELF file. The reason for this is that, as modern file systems and operating systems use the buffered IO method, and therefore, the place where disk writes are performed is a part of the *sync* operation rather than the *write* function.

The code that performs the *sync* operation in the kernel is specified in the member function pointer "`sync_fs`" of "`s_op`" (super-block operation), which is a field of the super_block structure. This "`sync_fs`" is a field set when the file system driver mounts a volume after the drivers is loaded. For this, file system driver files have an "`s_op`" structure, a "`sync_fs`", and the code of the function which will be assigned to the "`sync_fs`".

In the case of TFFS, the function `tffs_write_inode` is called from a specific branch within `tffs_sync_fs`. From the name of the function, we assumed that this serves to write the metadata and file contents in inodes on the RAM to the flash drive. In further analysis, we found it indeed perform such operation. And, from its code, we were able to identify the format and the location in which file creation time is recorded, which was our goal of this part of the analysis.

5. Results of file system analysis

5.1. Structure of the file system

The analyzed TFFS has a similarity to FAT-based file system. The TFFS divides a volume into a boot sector, a backup boot sector, a cluster allocation map (CAM) region, and a data region as shown in Fig. 4.

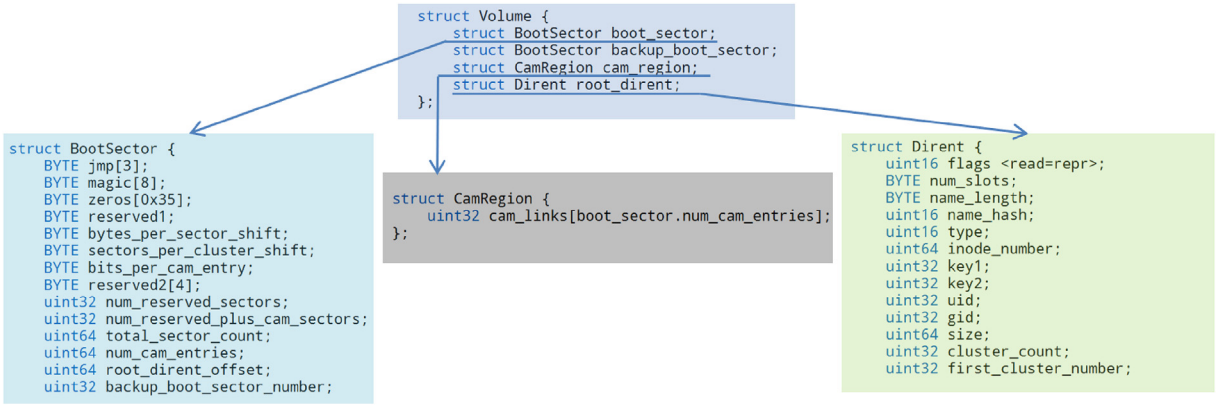

Fig. 4. Boot sector/CamRegion/Dirent entire information.

5.1.1. Boot sector

The volume information is recorded. The list of the field in the boot sector is shown in Fig. 4.

5.1.2. Backup boot sector

A copy of the boot sector is stored.

5.1.3. CAM region

The array of CAM entries is shown in Fig. 4. These functions are similar to the FAT entries of the FAT-based file system. Each CAM entry represents the allocation status of the corresponding cluster (usually 4 KB or 16 KB blocks) in the data region and the location of the next cluster in the cluster chain.

5.1.4. Data region

The data region is an array of equal-sized clusters and stores content of directories and files. The content of a directory is an array of directory entry (*dirent*) structures, one for each file and one for each subdirectory in that directory. The structure of dirent is shown in Fig. 4. For a file, on the other hand, its content is saved directly as a raw byte sequence. If the contents of a directory or file is too large to fit in one cluster, CAM entries are used to form a cluster chain and the content is recorded sequentially in each cluster constituting the chain.

5.2. Possibility of file system unit recovery

During deletion of a file, the file system driver does not deleted the whole content of the file, rather, it only marks the CAM entries corresponding to the file content as unallocated spaces. Therefore, information on deleted files can be analyzed by extracting unreachable areas through unallocated spaces and directory traversal. In the unallocated spaces extracted from the previous two cases, the video file was completely restored by the proposed method. We plan to pursue an analysis study according to the recovery rate and deletion method/DVRS recording period. Also, if most cluster chains are composed of adjacent clusters on the volume, carving is very effective and the file recovery rate can be high.

6. Case study

In this study, the file system analysis method was tested for DVRS of two models (HYUNDAI Sonata and KIA K7). In the case of the first car model (Sonata), the DVRS circuit board is damaged and the emergency download provided by the manufacturer is not available; therefore, only the chip-off method is possible. In the case of the second car model (K7), both the emergency download mode and chip-off method are possible with the circuit intact.

In both cases, eMMC memories of the same model shown in Table 2 is installed, and the obtained partitions are the same constitution described in Table 3. Among the obtained partitions, as shown in Fig. 3 and Table 3, nine partitions have “TFFS” written in their head parts.

Table 4 shows the number of files in those nine partitions analyzed by our reverse engineering method.

It shows that more files can be extracted from all TFFS partitions compared to the K7 DVRS's emergency downloads mode. Specifically, the “system” area where the operating system resides in and the “userdata” and “ydata” areas where the user settings and behavioral logs stored are partitions that cannot be extracted by the emergency download function. As shown in Fig. 5, the area where the normal video file is recorded, such as “Drive_Normal”, “Drive_Crash_Event”, and “Switch_Event” of the “yvideo” partition area of K7 DVRS, is extracted with the same result in both the modes.

However, the proposed method can additionally analyze user actions, system logs, and thumbnail images in “Database”, “Systemfile”, and “Thumb” folders. Moreover, in the system log, traces of user behavior are recorded in the form of text and log files. For example, you can check G-sensor-detected events with speed and gear lever position records. why the DVRS video file was created/

Table 4
List of major filesystems in DVRS.

Partition name	Emergency download	Our method (Sonata)	Our method (K7)
cache	0	0	0
persist	0	5	5
system	0	18860	20180
userdata	0	1764	1864
yapp	0	17	14
ybackup	0	0	0
ydata	0	3	3
yupdate	0	0	0
yvideo	446	1264	905

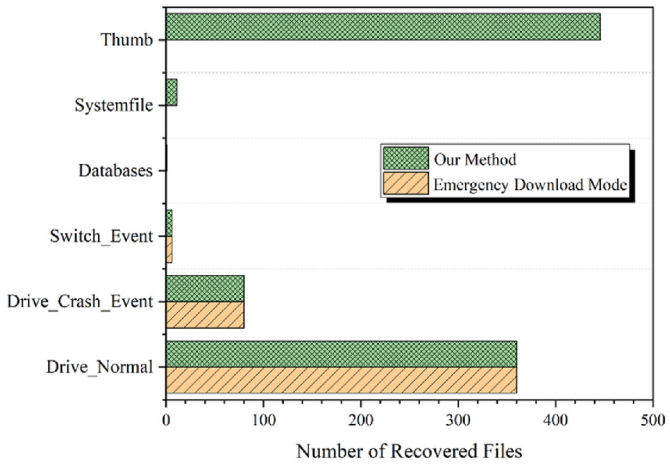

Fig. 5. Compare numbers of recovered files in "yvideo" partition.

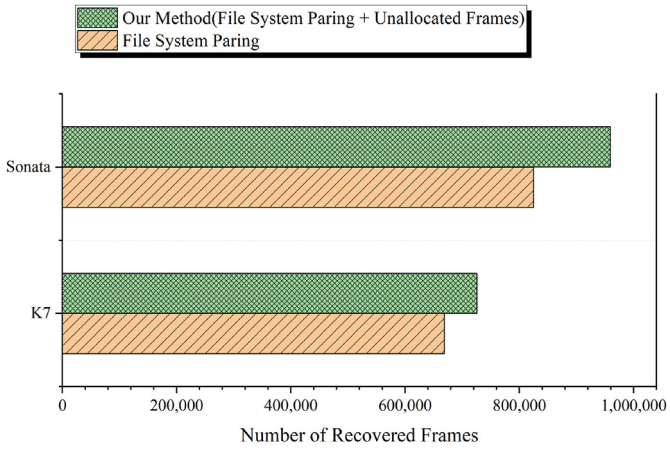

Fig. 6. Compare numbers of recovered frame in "yvideo" partition.

deleted, and what the connection passcode was when connecting to WiFi.

In addition, the unallocated area can be extracted by searching for clusters marked with deleted markers in the CAM region. An additional image frame can be reconstructed, as shown in Fig. 6, by using the frame-by-frame reconstruction technique on these unallocated areas.

Through this case analysis, we compared the method provided by the filesystem manufacturer with the method we developed and confirmed that the same file system analysis performance can be obtained. In addition, we showed that additional analysis is possible by collecting unallocated areas for partitions where image data might remain. For further information, please refer to the existing papers for analysis of the video files that are extracted in this frame-by-frame manner (Na et al., 2013).

7. Conclusion and future work

This paper describes the methodology and results of the file system analysis for operating systems operating in RT environments such as vehicles. This study was started because of the need to analyze a dedicated file system used in embedded devices that must ensure real-time performance. As embedded memories are installed in embedded devices and functions of the devices are diversified, it is necessary for forensic agencies to analyze the stored data. However, the acquisition process is difficult because of the characteristic of the memory. Even if the volume data is acquired in this manner, it is difficult to analyze the file units because of the dedicated file system. Although embedded device manufacturers are preparing a method to acquire the data even if the device is disconnected from the peripheral control device, this method cannot be used if the board is damaged. In addition, it is often not possible to obtain all data by the method provided by the manufacturer. Therefore, a digital forensic analysis might be able to analyze the file systems operating in the RT environment only by increasing the knowledge disclosed by the manufacturer and the kernel-level system understanding.

Moreover, it is expected that embedded systems in various RT environments (satellite, smart car, robot system) will be analyzed. We plan to analyze various dedicated file systems in the future.

Acknowledgements

This work was supported by National Forensic Service (NFS2023DTB04, NFS2023STR01), Ministry of the Interior and Safety, Republic of Korea.

References

Bmw, 2022. "BMW operating system 8. https://www.press.bmwgroup.com/global/article/detail/T0401875EN.

Buquerin, Klaus Gomez, Kevin, Corbett, Christopher, Hof, Hans-Joachim, 2021. A generalized approach to automotive forensics," Proceedings of DFRWS EU 2021, Forensic Science International: Digit. Invest. ume 36. https://doi.org/10.1016/j.fsidi.2021.301111.

Cotroneo, Domenico, De Simone, Luigi, Natella, Roberto, 2021. Timing covert channel analysis of the vxworks mils embedded hypervisor under the common criteria security certification. Comput. Secur. 106, 102307.

Daily, J., Johnson, J., Kongs, A., 2015. Vehicle electronic security and "hacking" your car," slide deck from SAE Texas meeting on car hacking. http://tucrrc.utulsa.edu/Publications/SAE%20Texas%20Meeting%20On%20Car%20Hacking%2016%20Jan%202014.pdf.

Govin, Brice, et al., 2015. Reverse engineering tool requirements for real time embedded systems. In: Proceed. SATToSE'15..

Hyundai, 2022a. Built-in cam 22. https://www.hyundai.com/kr/en/sedan/sonata/19fc/multimedia.

Hyundai, 2022b. "Connected car operating system,". https://tech.hyundaimotorgroup.com/mobility-service/connected-car-service/.

Kim, Younghun, Shin, Dongkun, 2015. Improving file system performance and reliability of car digital video recorders. IEEE Trans. Consumer Electron 61.2. 222–229.

Kopencova, Dagmar, Rak, Roman, 2020. Issues of Vehicle Digital Forensics," Proceedings of the XII International Science-Technical Conference AUTOMOTIVE SAFETY. IEEE.

Le-Khac, Nhien-An, et al., 2020. Smart vehicle forensics: challenges and case study. Future Generat. Comput. Syst. 109, 500–510.

Lee, K., Choi, J.H., Park, J., Lee, S., 2021a. Your car is recording: metadata-driven dashcam analysis system. In: Proceedings of DFRWS APAC 2021, Forensic Science International: Digital Investigation. https://doi.org/10.1016/j.fsidi.2021.301131.

Lee, Junghwan, Hyunchan, Park, ji-woo, Lee, Oc-Yeub, Jeon, 2021b. A case study on using car AVN system. Trans. KSAE. 29, 855–862.

Mercedes-Benz, 2022. "Mercedes-Benz operating system,". https://group.mercedes-benz.com/careers/about-us/mercedes-benz-operating-system/. https://web.archive.org/web/20220928100723/https://group.mercedes-benz.com/careers/about-us/mercedes-benz-operating-system/.

Na, Gi-Hyun, et al., 2013. Frame-based Recovery of Corrupted Video Files Using Video Codec Specifications. IEEE Transactions on Image Processing, pp. 517–526, 23.2.

Samsung, 2022. "eMMC5.1,". https://semiconductor.samsung.com/estorage/emmc/emmc-5-1/klmcg8gesd-b04q/.

Seo, Sukhyun, Kim, Junsu, Kim, Su Min, 2017. An analysis of embedded operating systems: Windows CE Linux VxWorks uC/OS-II and OSEK/VDX. Int. J. Appl. Eng. Res. 12.18. 7976–7981.

Tesla, 2022. "Dashcam and sentry mode". https://www.tesla.com/ownersmanual/model3/en_gb/GUID-3C7A4D8B-2904-4093-9841-35596A110DE7.html.

Tuxera, 2022. Embedded file systems. https://www.tuxera.com/products/.

Volkswagen, 2022. "VW.operating system. https://www.volkswagenag.com/en/news/fleet-customer/2021/01/transformers.html.

Zhang, Runyu, et al., 2020. Loffs: a low-overhead file system for large flash memory on embedded devices. In: Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). IEEE.