applied sciences

MDPI

Article

Android-Based Audio Video Navigation System Forensics: A Case Study

Haein Kang [1], Hojun Seong [2], Ilkyu Kim [3], Wookjae Jeong [4], Seong-Je Cho [4], Minkyu Park [5] and Sangchul Han [5,*]

[1] Department of AI-Based Convergence, Dankook University, Yongin 16890, Republic of Korea; hikang@dankook.ac.kr
[2] Department of Computer Science & Engineering, Dankook University, Yongin 16890, Republic of Korea; hohojun0930@dankook.ac.kr
[3] Department of Applied Computer Engineering, Dankook University, Yongin 16890, Republic of Korea; ik.kim@dankook.ac.kr
[4] Department of Software Science, Dankook University, Yongin 16890, Republic of Korea; wookjae@dankook.ac.kr (W.J.); sjcho@dankook.ac.kr (S.-J.C.)
[5] Department of Computer Engineering, Konkuk University, Chungju 27478, Republic of Korea; minkyup@kku.ac.kr
* Correspondence: schan@kku.ac.kr

Abstract: Vehicle digital forensics includes the process of collecting and analysing digital data stored in vehicles to find evidence related to traffic accidents or crime scenes. Through this process, we can reconstruct digital events by recognizing various information such as driver behaviour, driving patterns, vehicle destinations, and smartphones connected to a vehicle. Recently, many vehicle digital forensic studies have been conducted, but few of them have dealt with Android-based infotainment (or audio video navigation, AVN) systems. While many AVN systems adopt Android as the operating system, digital forensics of Android-based AVN systems is not easy. This is because Android-based AVN systems support various storage devices and data formats and have various data sources, making consistent data collection or analysis difficult. In this article, we perform digital forensics on four AVN systems: two Android 4.2.2 Jellybean-based and two Android 4.4.2 KitKat-based, aiming to develop a data acquisition and analysis method appropriate for each Android version. As a data collection method for the AVN system forensics, logical data acquisition is performed on the Jellybean-based systems and physical data acquisition on the KitKat-based systems. For the collected data, we identify and analyse Bluetooth data, navigation data, and system logs individually. Next, we investigate the differences in the storage structure and file location of major digital artefacts depending on the Android versions and show the limitations of the individual data analysis. Finally, we construct a timeline for the driver's activities by integrating and analysing the diverse artefacts which consist of Bluetooth data, navigation data, and system logs.

Keywords: vehicle digital forensics; infotainment system; AVN system; Android; autopsy

check for updates

Citation: Kang, H.; Seong, H.; Kim, I.; Jeong, W.; Cho, S.-J.; Park, M.; Han, S. Android-Based Audio Video Navigation System Forensics: A Case Study. *Appl. Sci.* **2023**, *13*, 6176. https://doi.org/10.3390/app13106176

Academic Editors: Arcangelo Castiglione, Stephan Neumann, Jurlind Budurushi and Malika Bendechache

Received: 31 March 2023
Revised: 30 April 2023
Accepted: 16 May 2023
Published: 18 May 2023

1. Introduction

A vehicle infotainment system is one of the devices that are built into vehicles. It provides drivers with enhanced convenience and safety. A driver or passengers can play music, navigate routes, and make hands-free calls [1,2]. Recent infotainment systems can work with smartphones and tablets, and we can use the apps installed on smartphones or tablets on the display of the infotainment system [3].

These systems store useful information for forensics in traffic accidents or crime situations [4] such as the vehicle's GPS log, control device records, speed and so on. Forensic experts can reconstruct events that occurred while driving from the stored data. The events include call logs, text messages, apps used, and so on [5,6]. Forensic findings can be matched against statements about accidents or crimes. The results can also be used to determine the cause of an accident or to identify a suspect.

A digital forensic tools can extract and analyse data stored in the infotainment system of a vehicle. For example, Berla Corporation, a vehicle forensic company, released a vehicle forensic toolkit iVe. It provides a tool to extract and analyse various vehicle data such as driving records, location information, phone call records, music playback records, etc. [7,8]. The iVe supports data acquisition from the infotainment and telematic systems of over 4600 vehicle models. However, it does not support data acquisition from Android-based infotainment systems [9,10]. Through several experiments, we found that it is difficult to conduct a digital forensic investigation on an Android-based infotainment system.

- Android-based infotainment systems may have a different operating system version and hardware configuration. It is necessary to identify the characteristics of a target infotainment system to collect and analyse data of the system.
- Android-based infotainment systems use various data formats for driving records, music playback records, radio reception records, and so on.
- In Android-based infotainment systems, there are various sources of data such as Bluetooth, navigation, system logs, OBD-II, telematics, Wi-Fi connection, Android apps, etc.
- There is a lack of tools to perform digital forensic investigations of Android-based vehicle infotainment systems.
- Android-based infotainment systems have different data storage methods depending on each Android version. We also found that a data protection technique was applied in some versions [11].
- Therefore, it is difficult to make consistent forensic data collection as well as forensic analysis for Android-based infotainment systems.

There is an abundance of research on infotainment system forensics [3,5,6,8–10,12]; however, only a few study Android-based vehicle infotainment systems. In particular, to the best of our knowledge, there has been no research collecting data from various sources, including Android operating systems, to construct a timeline of user activities. Furthermore, previous research does not analyse various data sources; hence, the available artefacts are limited.

In this article, a digital forensic investigation is performed on four Android-based infotainment systems, also known as audio video navigation (AVN) systems [13]: two Android 4.2.2 Jellybean-based and two Android 4.4.2 KitKat-based. To collect forensic artefacts from the AVN systems, a logical acquisition method is applied to the Jellybean-based AVN systems and a physical acquisition method to the KitKat-based AVN systems. For the collected artefacts, Bluetooth data, navigation data, and system logs are identified and analysed individually. Next, we examine the differences in the storage structure and file location of major digital artefacts depending on the Android version and show the limitations of the individual data analyses. Finally, we construct a timeline of the driver's activities by integrating and analysing the various artefacts containing Bluetooth data, navigation data, and system logs.

The main contributions of this work are as follow.

- This work provides the methodology for Android-based infotainment system forensics. Depending on the version of the Android operating system, the method of extracting data from the storage device, logically or physically, is determined.
- Data are obtained from various sources including Bluetooth, navigation and system logs. Many kinds of artefacts are collected and analysed.
- By integrating the analysis results, a timeline of user activities is constructed. A timeline can help an investigator infer the user's behaviour or cause of the accident.
- In particular, the data obtained from the system logs provide temporal information and vehicle events, useful to construct a timeline of user activity.

The structure of this article is as follows. In Section 2, we analyse the recent trends and problems in digital vehicle forensics. We describe the characteristics of the four target infotainment systems and the forensic procedures in Section 3. In Section 4, we describe the forensic results. Section 5 discusses the analysis of the forensic results. We conclude our work in Section 6.

2. Related Work

The vehicle infotainment system is a compound word of "information" and "entertainment". Hyundai Motors and Kia Corporation, Korean multinational automotive manufacturers, sometimes use the term AVN (audio video navigation). Users use navigation or telematic functions and check various statuses of the vehicle. Users can also enjoy entertainment using audio, radio, and DMB applications. Most infotainment systems adopt Android, Linux, QNX, or Windows as the operating system to support existing applications [14–17].

Bequerin et al. [18] proposed the four-phase forensic process for vehicles. The process consists of forensic readiness, data acquisition, data analysis, and documentation. The forensic expert first selects a forensic method and the data (forensic readiness). The forensic expert acquires data (data acquisition) and interprets the acquired data (data analysis). The forensic expert completes the forensics by writing a final report (documentation). The authors show the process is cost-effective by inspecting an electric vehicle.

Lacroix et al. [5] described the components of modern vehicles and discussed the possibilities for vehicle digital forensics. By performing forensics on a Ford F-150 truck SYNC infotainment system, they showed that forensics is possible. The authors also emphasized the development of appropriate forensic tools, frameworks, and methodologies. These will help law enforcement agencies identify suspects.

Whelan et al. [8] performed forensics on the infotainment systems of a Dodge Dart and a Toyota Highlander Limited using the iVe. These vehicles are equipped with infotainment and telematic systems made by the Berla Corporation. iVe is an evidence-gathering and analysis tool. They extracted navigation data and user-related events (door open, WiFi and Bluetooth connection, recent location, track log), including the serial number, part number and vehicle identification number (VIN). Scientific Working Group on Digital Evidence (SWGDE) guideline [19] classifies these data as artefacts.

Le-Khac et al. [9] noted that vehicle data forensics was not sufficiently studied and explained the challenges associated with the digital investigation of vehicle systems: (1) A forensically sound approach is needed for vehicle investigation. (2) It is also necessary to design tools that help the collection and analysis of data from vehicles. They identified the types of forensic artefacts recoverable from vehicles. They discussed the hardware and software solutions to collect and analyse the vehicle's artefacts. Using a Volkswagen, an Audi, and a BMW as case studies, they demonstrated that forensic-related data could be recovered from memory chips and navigation devices.

Bortles et al. [6] selected a 2015 model Ford F-350 Super Duty 4 × 4 pickup as a forensic case study and conducted a forensic examination. They collected and analysed the data recorded by the infotainment and telematic modules. The examined data elements contained door open/close events, gear shift events, parking light on/off events, phone calls, short message service (SMS), and vehicle tracklogs. According to their findings, there seems to be a limited amount of data which the vehicle can record, so not all vehicle events, phone events, and tracklogs are logged.

Lacroix [10] tried to find out what kinds of information are recorded by vehicle infotainment systems, how it is collected, and what the collected data implies. Lacroix acquired many datasets from various infotainment systems from different platforms. However, there was variance across the types and quantity of the acquired data. It was not easy to acquire file images of the infotainment systems because they were tightly coupled with different software, hardware, and builds. Furthermore, there was a lack of standardization across the OEMs. Lacroix's study pointed out that a third-party solution such as Berla iVe may work for very specific makers and models of vehicles.

Jacobs et al. [12] collected and analysed the data of an entertainment system on a Volkswagen Golf version 6, 2012 station wagon. By scanning the car diagnostic system using VCDS, a vehicle diagnostics software from RossTech, they acquired the following information: (a) data and time used by the examination computer, (b) VIN/chassis number of the car, (c) modules scanned by the software, (d) repair order, etc. They also extracted data from a type 510 radio navigation system (RNS) using either JTAG or chip-off. They analysed the data on the hard drive from the RNS-510 using the FTK Imager forensic tool and found two partitions: a Windows FAT32 and a WindRiver Systems DocFs 2.0. The FAT32 partition included navigation map information containing location data. They performed file carving on the WinRiver partition using the Photorec tool and recovered multiple mp3 files.

Mansor et al. [20] proposed a mobile application, DiaLOG, which makes certain that only an authorized mobile device can be connected to a target car through a secure protocol. The application collects data while protecting users' privacy. The mobile device is connected to the car using a WiFi connection through an on-board router. DiaLOG can record the car's diagnostic data, such as DTCs (diagnostic transmission codes), ECU content, interface connection to the car, crash-like data, and the normal frequency of messages through the CAN bus.

We carried out a preliminary forensic examination on an Android 4.2.2 Jellybean-based AVN system embedded in a Kia NIRO EV [11]. We collected and examined navigation-related files using a logical acquisition method, and identified user data such as start log, favourite routes, last destination search history, etc. This article is an extension of a preliminary study.

3. Forensic Process and Target System

We conducted a digital forensic investigation of four Android-based AVN systems to collect and analyse app and log data. This section explains the process of the forensic investigation and describes the characteristics of the target systems and their components.

3.1. Forensic Investigation Process

Figure 1 illustrates the process of our forensic investigation of Android-based AVN systems. Our study conforms to the automotive forensic process proposed in [18]. In the forensic readiness phase, the Android version of the AVN system is identified. Then a method to image the data partition of the AVN system is chosen since a cost-efficient data acquisition method may differ depending on the Android version. In the data acquisition phase, we can utilize the USB interface and 'dd' command to extract data logically, or utilize the chip-off technique and forensic-imaging tools, such as Autopsy or FTK Imager, to extract data physically. We can divide the data analysis phase into two parts: analysis of individual source and integration. In the former part, we analyse user behaviour with an individual data source: Bluetooth app, navigation app and system log. In the latter, we integrate the analysis results and create a timeline.

3.2. Digital Forensic Tools

Digital forensic tools are HW/SW tools that forensic analysers use to collect data and recover digital evidence from digital devices. There are many forensic tools used by law enforcement agencies, government organizations, companies, etc. Some of them are listed in the following.

- EnCase [21]: a forensic investigation tool that mainly examines digital evidence on hard drives and various storage media. The evidence collected by EnCase has been used in many court cases.
- FTK (Forensic Toolkit) [22]: another forensic tool that has been cited by the government and the court. Its functionality is similar to EnCase. There is an associated software FTK Imager that can generate an image file of a hard drive.

- Autopsy [23]: a free and open-source digital forensic tool. It supports various file systems, archive files, email formats and contact files. It also provides plugin functionality. Users can perform a customized analysis by creating or installing plugins.
- X-Ways Forensics [24]: a computer forensic examiner. Its functionality is similar to EnCase and FTK, but it does not support some functions such as network connection analysis or remote capture.
- iVe [7]: a vehicle forensic tool. It supports various vehicles such as passenger cars, trucks and motorcycles. It provides integrated analysis of various systems such as infotainment, telematics and safety systems.
- Andriller [25]: a free and open-source forensic tool for Android smartphones. It can automatically extract data from smartphones, parse folder structure, decode backup and database files, crack lockscreens, and so on.

Autopsy was used to analyse a disk image in our research. The reason why Autopsy was chosen is as follows. First, Autopsy is a free and open-source software. Most forensic tools are not free and their license price is very high. Second, Autopsy's functionality can be customized. For example, we deployed the keyword search ingest plugin in order to index files found in the disk image. Third, Autopsy has been used for the forensic analysis of hard disk or file systems and data recovery in many fields [26–28]. Beside Autopsy, several forensic tools are used in our research, as shown in Table 1.

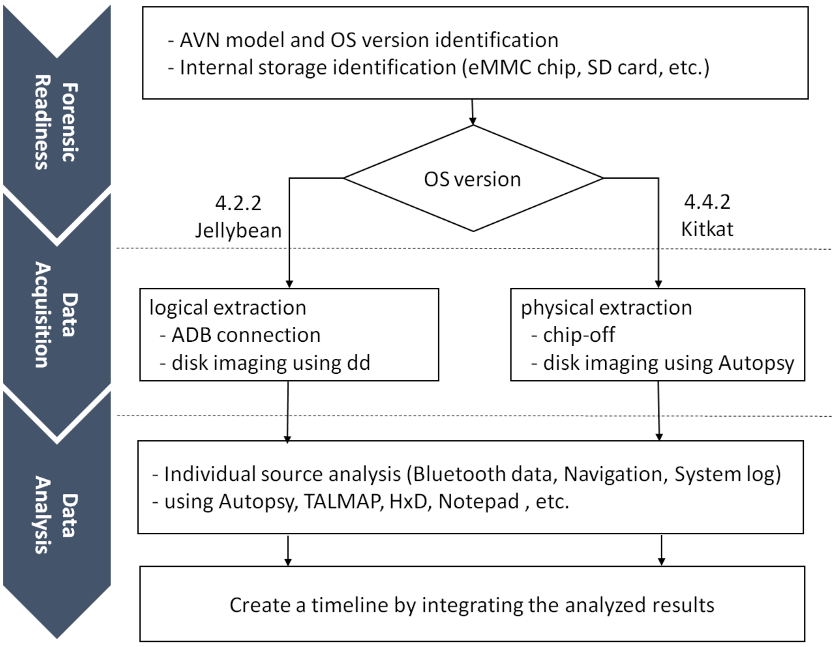

Forensic Readiness
Data Acquisition
Data Analysis
- AVN model and OS version identification
- Internal storage identification (eMMC chip, SD card, etc.)
OS version
4.2.2 Jellybean
4.4.2 Kitkat
logical extraction
- ADB connection
- disk imaging using dd
physical extraction
- chip-off
- disk imaging using Autopsy
- Individual source analysis (Bluetooth data, Navigation, System log)
- using Autopsy, TALMAP, HxD, Notepad , etc.
Create a timeline by integrating the analyzed results

Figure 1. Process of Android-based AVN system forensics.

3.3. Target Systems

The target AVN systems used in our experiments have two storage devices: eMMC (embedded multimedia card) chip and SD (secure digital) card. The two storage devices are different in installation method and use in the AVN systems. The eMMC chip is embedded on the PCB (printed circuit board) and contains the user data of the Android system, while the SD card is inserted into a slot and contains data related to navigation updates. Since we are interested in the user data generated and stored by the AVN system a driver or passenger manipulates, the target storage device is the eMMC chip. We physically identify the eMMC chip on the PCB and logically locate the user data in the file system when the device has been successfully imaged.

Table 2 shows the features of the target AVN systems. The vehicle's model name, the AVN system's manufacturer, model number, and Android OS version are presented. Two of them are based on Android 4.2.2 Jellybean and the others are based on Android

4.4.2 KitKat. The file system type of storage device is ext4. Bluetooth, navigation, telematics and radio apps are commonly installed.

Table 1. Forensic tools.

Tool	Description	Use
Autopsy	a free and open-source hard drive investigation tool	to analyse hard drive images of an AVN's internal partition
ADB (android debug bridge)	a debugging tool for Android-based devices that connects a PC with the Android device	to enter the AVN system and identify the disk partitions
dd (disk dump)	a command-line utility that converts and copies (device) files	to image and copy the AVN's internal partition to an SD card
DB4S (DB Browser for SQLite)	a tool for manipulating SQLite-compatible database files	to identify tables and fields and search for keywords in the database files
HxD	a hex editor that edits binary files	to analyse binary files generated by the navigation software with the file extension .dat or .bin
Notepad	a simple text editor that edits various types of text files, including HTML and XML, and supports various encodings	to read and analyse log files
Talmap	a Korean navigation software	to convert Talmap's coordination information to GPS information
Epoch converter	a simple program that converts Unix epochs to human readable date and time	to convert epochs found in file names to local time (KST)

Table 2. Target AVN systems.

Vehicle Model	AVN Model	Android Version	Linux Kernel	File System	eMMC Chip
Kia K5 (2015)	LG Electronics LAN5020KKJF	4.2.2 Jellybean	3.1.10	ext4	Micron MTFC4GACA AAM-4M IT (32 GB)
Kia NIRO EV (2018)	LG Electronics IA88431DELE	4.2.2 Jellybean	3.1.10	ext4	Micron MTFC4GACA JCN-4M IT (32 GB)
Hyundai Sonata DN8 (2019)	Hyundai MOVIS 96560L1070SS	4.4.2 KitKat	3.18.24	ext4	Samsung KLMCG8G ESD-B03Q (64 GB)
Kia All New Morning (2020)	LG Electronics 965601Y000MB2	4.4.2 KitKat	3.18.24	ext4	Samsung KLM4G1FE PD-B031 (64 GB)

ADB (android debug bridge) is a debugging tool for Android-based systems. If ADB of an Android-based system is activated, a PC can be connected to the system over USB and used as a terminal or emulator to interact with the system or enter the shell. We can create an image of the system's storage device using a shell command. However, Android-based systems are usually delivered with ADB deactivated and we cannot activate ADB in the normal user mode. To do so, we need to enter the engineering mode, which is a hidden option in which developers or engineers can perform advanced functions to test or A/S the system.

To activate ADB on the target AVN systems, we need to enter the engineering mode. We reverse engineered the target systems and discovered how to enter the engineering mode. We do not describe this process in detail since reverse engineering is beyond the scope of this article. We also found that whether ADB can be activated or not depends on the Android OS version. To be specific, ADB can be activated in Android 4.2.2 Jellybean but not in Android 4.4.2 KitKat. If ADB can be activated, we have a choice of data extraction methods: logical or physical. Note that the cost of physical extraction is much higher than logical extraction. Thus, in the forensic readiness phase, a cost-efficient method for imaging AVN systems can be chosen considering the Android OS version.

4. Android-Based AVN System Forensics

This section describes the procedure for extracting and analysing data from Android-based AVN systems. After the Android OS version has been identified, the data is extracted logically or physically. Then Bluetooth data, navigation data and system logs are analysed. The artefacts found in each data source are identified and located. AVN systems from a Kia K5 (2015) and a Kia NIRO EV (2018) are investigated in Section 4.1 and a Sonata DN8 (2019) and a Kia All New Morning (2020) in Section 4.2.

4.1. Android 4.2.2-Based AVN System Forensics

First, the AVN systems are removed from the vehicle, and their manufacturer and model number are identified. Then, we discover their Android OS version and other features utilizing the manufacturer's documents and navigation update service [29,30]. For example, the AVN system removed from a Kia K5(2015) was physically identified as LG Electronics' LAN5020KKJF. We searched for the related documents and navigation update program. Finally the AVN system's OS was identified as Android 4.2.2 Jellybean and we obtained its detailed specification.

Applying the results from Section 3.3, we entered the engineering mode and found the hidden menu to activate ADB. We connected a PC to the target AVN system through a USB interface, activated ADB, and configured the ADB-related settings. Then, we successfully entered the ADB shell. However, to explore the file systems of the target system, the root privilege is required. To obtain root privilege, we exploited the dirty COW (copy-on-write) vulnerability—a privilege escalation vulnerability—of the Linux kernel [31]. Now the data in the storage devices could be accessed and identified. This procedure is applicable to the two Android 4.2.2 Jellybean-based AVN systems.

Both AVN systems have block device files `mmcblk0` and `mmcblk1`. File `mmcblk0` represents the storage device eMMC chip and has partitions such as `/data`, `/system`, `/ivilog` and `/ivibackup`. Particularly, in the subdirectories of `/data`, many files that seem to be user data are found. File `mmcblk1` represents the storage device SD card and contains the files for the map data of the navigation app. Since we are interested in user behaviour, we created an image of `mmcblk0` using command dd. Figure 2 illustrates the above procedure.

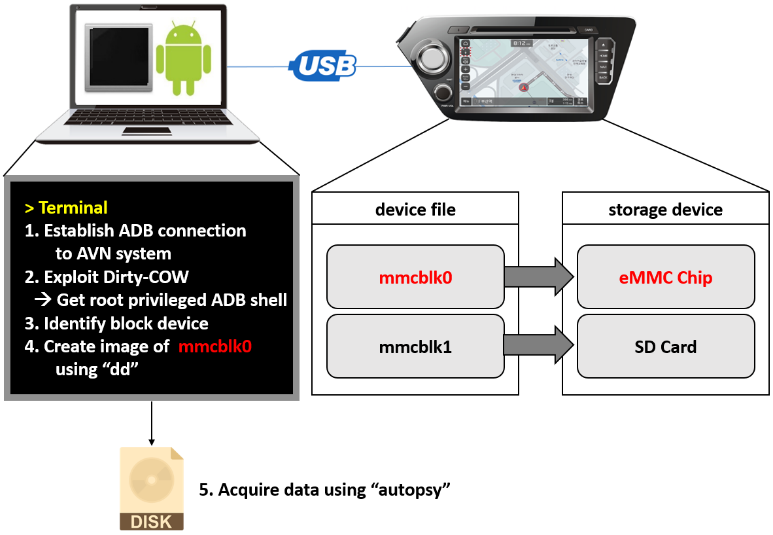

USB
> Terminal
1. Establish ADB connection to AVN system
2. Exploit Dirty-COW → Get root privileged ADB shell
3. Identify block device
4. Create image of mmcblk0 using "dd"
device file
mmcblk0
mmcblk1
storage device
eMMC Chip
SD Card
DISK
5. Acquire data using "autopsy"

Figure 2. Data acquisition from Android 4.2.2-based AVN systems.

We collected directories and files from the partitions in the created image utilizing Autopsy. In `/system`, APK (Android application package) files installed on the system are found. They are related to the functions provided by the AVN system such as Bluetooth communication, navigation, telematics and radio. These apps create user data in partition `/data` while running. In the following sections, Bluetooth data, navigation data and Android system logs are analysed and many artefacts are identified.

4.1.1. Bluetooth Data

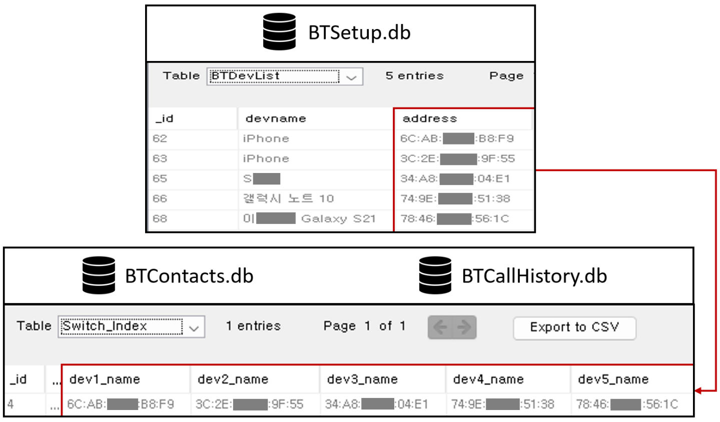
If passengers connect a mobile phone to the AVN system via Bluetooth, they can use the hands-free function of the AVN system to receive and send phone calls. In this case, Bluetooth-related apps record the data about the connected devices and phone calls. Bluetooth-related apps, such as Blutooth.apk, BTMedia.apk, BTService.apk, BTSetup.apk and BTPhone.apk, are located in the subdirectories of `/system`. The data recorded by these apps is stored as database files in `/data/data/com.android.providers.bluetooth/database/`. Table 3 shows the database file, table and attribute names of the main artefacts. `BTSetup.db` stores the information about the connected devices. `BTContacts.db` and `BTCallHistory.db` store the contact lists and phone call history of each connected mobile phones, respectively.

Table 3. Bluetooth data of the Jellybean-based AVN systems.

File Name	Table	Attribute	Artifact
BTSetup.db	BTDevList	devname	device name
		address	MAC
	Switch_Index	dev#_name	MAC
BTContacts.db	Switch_Index	dev#_name	MAC
	Dev#Contacts	name	given name
		fname	family name
		num#	phone number
BTCallHistory.db	Switch_Index	dev#_name	MAC
	Dev#CallHistory	type	dialed | received | missed
		name	name
		fname	fname
		number	phone number
		tel_type	[cell, home, other]
		date_time	call date

The main Bluetooth data artefacts are the MAC (media access control) address of the connected mobile devices, device name, contacts and recent call history. The MAC address of the connected mobile device can be found in `BTSetup.db`, `BTFavorites.db`, `BTContacts.db` and `BTCallHistory.db`. The target AVN systems can establish a total of five concurrent Bluetooth connections. We verified that the number of tuples of the `BTDevList` table is limited to five. As shown in Figure 3, the Bluetooth MAC address of the connected mobile device is stored as attribute `dev#_name` in the `Switch_Index` table of `BTContacts.db` and `BTCallHistory.db`, where '#' is the index of the MAC address in the`BTDevList` table in `BTSetup.db`. For example, the MAC address of the third mobile device in the `BTDevList` table of `BTSetup.db` is also stored in attribute `dev3_name` in the `Switch_Index` table of `BTCallHistory.db`.

A user can provide a name to their mobile phone. This name is broadcast to neighbouring devices via Bluetooth or WiFi. The target AVN systems store this name in attribute `devname` in the `BTDevList` table of `BTSetup.db`. The contact list of the mobile phone can be found in `BTContacts.db` which contains the name and phone number of each contact. The call history (log) can be found in `BTCallHistory.db`. It contains the call type, the name and phone number of the party retrieved from the contact list and the date and time of the call. The call type is denoted as `Dialled` (outgoing), `Received` (incoming) or `Missed`.



BTSetup.db

Table BTDevList 5 entries Page

_id	devname	address
62	iPhone	6C:AB:[redacted]:B8:F9
63	iPhone	3C:2E:[redacted]:9F:55
65	S[redacted]	34:A8:[redacted]:04:E1
66	갤럭시 노트 10	74:9E:[redacted]:51:38
68	이[redacted] Galaxy S21	78:46:[redacted]:56:1C

BTContacts.db BTCallHistory.db

Table Switch_Index 1 entries Page 1 of 1 Export to CSV

_id	...	dev1_name	dev2_name	dev3_name	dev4_name	dev5_name
4	...	6C:AB:[redacted]:B8:F9	3C:2E:[redacted]:9F:55	34:A8:[redacted]:04:E1	74:9E:[redacted]:51:38	78:46:[redacted]:56:1C

Figure 3. MAC address of the mobile devices connected via Bluetooth.

4.1.2. Navigation Data

There are many navigation software installed on infotainment or AVN systems. Their route-finding methods are based on various techniques such as Dijkstra, A*, and neural networks [32–34]. However, the artefacts generated by the navigation software such as favourite location, destination history and search history are similar. The navigation app installed on the target AVN systems is Talmap [35], a Korean navigation software. Its APK file is HKMC_Navi_ECN.apk. It generates data in the subdirectories of `/data/data/com.mnsoft.navi/` in various file formats such as database (.db), binary (.dat, .bin) and text (.txt).

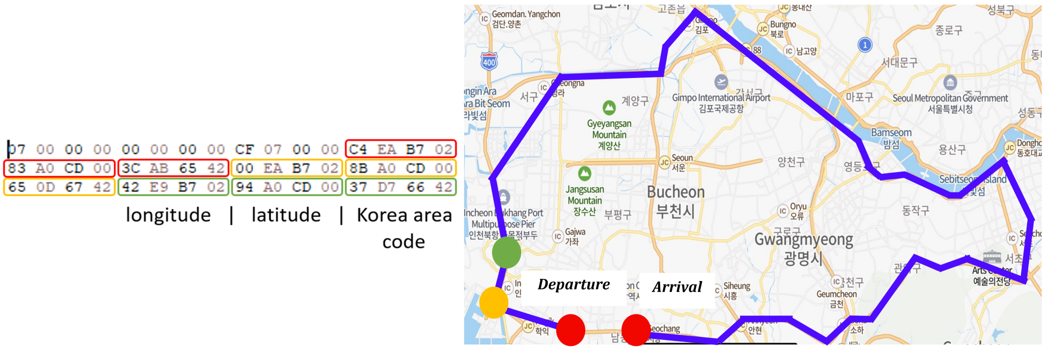
Table 4 lists the artefacts found in the database. The main artefacts found in the database include the recent location, search history, favourite location, and registered location. `Current_Loc_Table` stores the last location. It contains the coordinate information, address and administrative region code. The coordinate information is expressed in Talmap's own way. This information can be converted to the longitude and latitude using Talmap's service [35]. Location search history can be found in `Destination_Table` and `NonSearch_Table`. `Destination_Table` stores the last search result and `NonSearch_Table` stores several search results including the last one. These tables contain search words, destination addresses, distance/time to destination and coordinate information. `Memory_point_Table` and `RegisPnt_Special_Table` store the locations registered by the user. They contain the name, address and coordinate information of the registered locations.

Table 5 shows the navigation app data stored in non-database files. The artefacts found in the files include the start time of the AVN system, the destination search history, the last search record, GPS records and registered locations. Directory `UserData/KOR` contains six files all of which are related to the navigation app. File `Last_Route_info`, `USERPOI` and `USERRECENT` have the same artefacts found in the database file. File `startlog_[epochtime].txt` stores the navigation app logs when the AVN system boots and the navigation app starts. Its filename contains an epoch time which can be converted to a human-readable date/time using an epoch converter. We can infer when the engine starts from the date/time. File `FavoriateDest.bin` stores the destination search history similar to `Destination_Table` in `Navi_vr.db`. Particularly, it also stores the time when a user performs the search. File `Last_Route_info` stores the last search words and the coordinate of the destination. This file is only found in the LG Electronics LAN5020KKJF model.

File `GPSTrack.dat` records the GPS information. As shown in Figure 4 (left), the coordinate information and administrative area code for each point are stored as little-endian binary numbers. For example, the coordinate information of the departure point (in red rectangles) is 45607602 (02 B7 EA C4) and 13475791 (83 A0 CD 00). Furthermore, the administrative area code of the point is 1113959228 (3C AB 65 42). The coordinate information can be converted to the longitude and latitude (GPS) using Talmap's service. We convert the coordinate information of the points and reconstruct the position information of the car, as shown in Figure 4 (right). The coordinate information of the last point in `GPSTrack.dat` coincides with the coordinate information found in `CurrentLoc_Table` in `Navi_vr.db`. We can track the route of the car based on the analysis of `GPSTrack.dat`.

Table 4. Navigation data contained in the database file in Jellybean-based AVN systems.

File Name	Table	Attribute	Description
database/ Navi_vr.db	Current_Loc_Table	Info_Longitude	last longitude
		Info_Latitude	last latitude
		Info_CurAreaName	last Korea area code
	Destination_Table	Info_Name	last search word
		Info_Address	address of last destination
		Info_TotalDist	distance to last destination
		Info_TotalTime	time to last destination
		Info_Longitude	longitude of last destination
		Info_Latitude	latitude of last destination
	Memory_point_Table	Info_Name	name of favorite location
		Info_Address	address of favourite location
	RegisPnt_Special_Table	Info_Name	name of registered location
		Info_Address	address of registered location
		Info_Longitude	longitude of registered location
		Info_Latitude	latitude of registered location
	NonSearch_Table	Info_Name	search word
		Info_Address	address of destination
		Info_Tel	phone number of destination
		Info_Distance	distance to destination
		Info_Direction	time to destination
		Info_Logitude	longitude of destination
		Info_Latitude	latitude of destination



```
07 00 00 00 00 00 00 00 CF 07 00 00 C4 EA B7 02
83 A0 CD 00 3C AB 65 42 00 EA B7 02 8B A0 CD 00
65 0D 67 42 42 E9 B7 02 94 A0 CD 00 37 D7 66 42
```

Figure 4. GPS Tracking Results.

4.1.3. System Log

In the target AVN system, the system log is stored as text files in the directory `/ivilog /dropbox/`, whose file name contains the epoch time. Bluetooth connection logs and GPS information are recorded in the files. Table 6 shows an example of system logs generated when a mobile device connects to the AVN system. Class bt-hci generated HCI (host–controller interface) snoop logs at 12:44:29. Bluetooth HCI snoop logs record the events of Bluetooth communication for Bluetooth-related debugging in Android systems [36].

The logs include the information of Bluetooth packets exchanged between devices, MAC address of Bluetooth devices and metadata related to Bluetooth communication.

Table 5. Navigation data in non-database files in Jellybean-based AVN systems.

Location	File Name	Description
UserData/KOR/	startlog_[epochtime].txt	engine start time
	FavoriteDest.bin	destination information
		destination address
		destination coordinates
		administrative area code
		date/time of search
	Last_Route_info	last search word
		longitude/latitude of last destination
	GPSTrack.dat	GPS record (logitude, latitude, administrative area code)
	USERPOI	name of registered location
		address of registered location
		longitude/latitude of registered location
	USERRECENT	last search word
		address of last destination
		phone number of last destination
		longitude/latitude of last destination
		administrative area code

Table 6. Example of an HCI snoop log.

Time	Level/Class	Info
12:44:29.783	I/bt-hci	SENT Command to HCI. HCI_Create_Connection (Hex:0x0405 Param: 13) Ctrl (0)
		Parameters
		BD_ADDR of remote device: 74-XX-XX-XX-XX-XX
		Packet Types Supported: 0x0018 (2-DH1 3-DH1 DM1 DH1 2-DH3 3-DH3 2-DH5 3-DH5)
		Page Scan Repetition Mode: 0 (0x00)
		Page Scan Mode(BT1.1)/Reserved(BT1.2): 0 (0x00)
		Clock Offset: 0 (0x0000)
		Allow Role Switch: 1 (0x01)
12:44:29.786	I/bt-hci	RCVD Event from HCI. Name: HCI_Command _Status (Hex Code: 0x0f Param Len : 4) Ctrl(0)
		Parameters
		Status: Success (0x00)
		Num HCI Cmd Packets: 1 (0x01)
		Cmd Code: 0x0405 (HCI_Create_Connection)

In most Android-based systems, an HCI snoop log is stored in a binary format. In the target AVN systems, fortunately, the logs were decoded and stored in a text format, making it easy for us to identify and analyse the contents. Utilizing the system log data, we identified detailed information such as the Bluetooth connection time that we cannot obtain from the Bluetooth app data.

4.2. Android 4.4.2-Based AVN System Forensics

In this section, we extract and analyse the information of the AVN system from the Hyundai Sonata DN8 (2018) and Kia All New Morning (2020). The OS of the AVN systems is the Android 4.4.2 KitKat. We found a method of entering the system's engineering mode and hidden menu related to the ADB. However, we could not figure out the next steps to connect the AVN system to a PC and enable ADB communication. Therefore, logical data acquisition from the AVN systems was impossible and a chip-off method was used to obtain the data as a hardware-based data acquisition approach.

Using the chip-off method, we desoldered an eMMC memory chip from the PCB of each AVN system (Figure 5) and bridged the chip to collect the data. Of the two memory chips, one was from Micro and the other from Samsung. In a digital forensic process, because it is important to minimize damage to the chip to prevent data loss, we were very careful not to apply excessive heat during the chip-off procedure. The separated chip was placed in an eMMC card reader attached to a PC, and then its disk image was acquired using the Autopsy tool. Using the disk image, we analysed Bluetooth and navigation data, and then the system logs. We briefly explain the same artefacts found in the Jellybean-based systems.

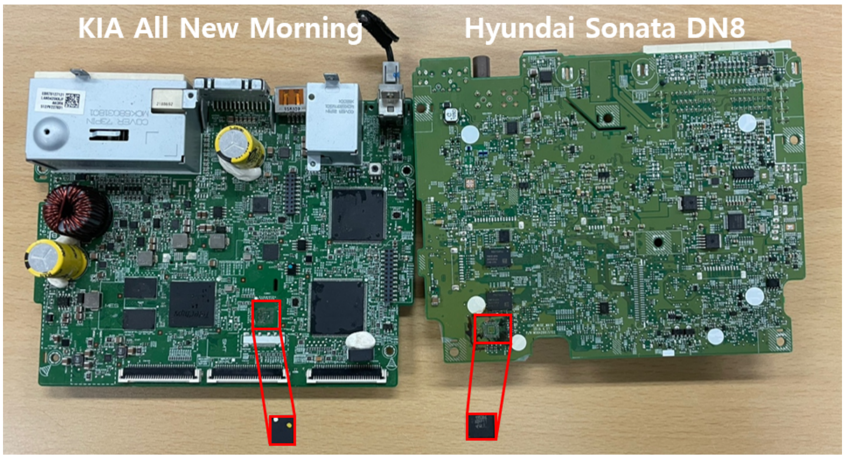


Figure 5. Disassembled AVN systems and desoldered eMMC memory chips.

4.2.1. Bluetooth Data

Similar to the Jellybean-based AVN systems, the KitKat-based AVN systems have records of the driver or passenger's mobile device connected to the AVN system via Bluetooth. The records are created by `Bluetooth.apk`, `BTMedia.apk`, `BTService.apk`, `BTSetup.apk`, and `BTPhone.apk`, and are stored in the database file (.db) in the `/data/data/com.android.providers.contacts/database/` directory. Among the records, the main forensic artefacts are the MAC address of the smartphone connected to the AVN system via Bluetooth, the smartphone name, the phone book, and the recent phone call history. Table 7 shows the file names, file paths, table names, and attribute names of key artefacts related to Bluetooth apps on the KitKat-based AVN system.

The Bluetooth MAC address of the smartphone paired with the AVN system via Bluetooth is recorded in the `BluetoothContacts.db` and `contacts#.db` files. The name of the smartphone is recorded in the `friendly_name` attribute in the `Bluetooth_devices` table of the `BluetoothContacts.db` file. The phonebook and favourite contacts are stored in the `MC_[Bluetooth MAC address].db` (e.g., `MC_948bc188bab3 .db`) and `MF_[Bluetooth MAC address].db` (e.g., `MF_948bc188bab3.db`), respectively. They include the given name (first name), family name (surname), and phone number of each person in the phonebook or favourite contacts. The call history is stored in the `Calls` table of the `contacts#.db` file.

The table can records the phone number of a caller or receiver, and the time of the call. Through the `account_name` attribute of the table, the MAC address of the smartphone found in the `BluetoothContacts.db` file can be identified (see Figure 6). As a result, even if the AVN system is connected to several mobile devices, it is possible to identify a mobile device that has left a specific phone call log.

Table 7. Bluetooth data in KitKat-based AVN systems.

File Name	Table Name	Attribute Name	Description
BluetoothContacts.db	Bluetooth_devices	bd_address	MAC address of smartphone
		friendly_name	Smartphone's name
MC_[MAC address].db	Bluetooth_contacts	given_name	given name
		family_name	family name
		phone#_number	Phone number
MF_[MAC address].db	Favorites	display_name	English name
		display_name_alt	Korea name
		Number	Phone number
contacts#.db	Calls	Name	given name
		Number	Phone number
		Date	call date
		account_name	MAC address of smartphone

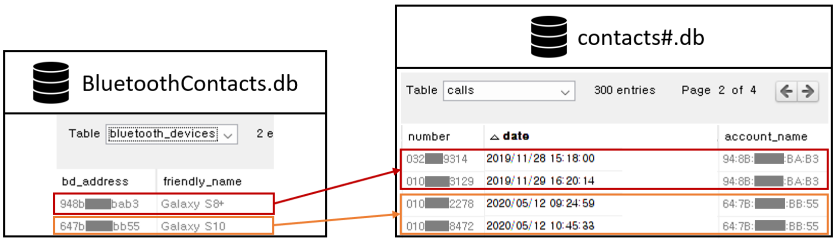


Table bluetooth_devices 2 e

bd_address	friendly_name
948b[redacted]bab3	Galaxy S8+
647b[redacted]bb55	Galaxy S10

Table calls 300 entries Page 2 of 4

number	date	account_name
032[redacted]9314	2019/11/28 15:18:00	94:8B:[redacted]:BA:B3
010[redacted]3129	2019/11/29 16:20:14	94:8B:[redacted]:BA:B3
010[redacted]2278	2020/05/12 09:24:59	64:7B:[redacted]:BB:55
010[redacted]8472	2020/05/12 10:45:33	64:7B:[redacted]:BB:55

Figure 6. Analysis of the `contacts#.db` file.

4.2.2. Navigation Data

Similar to the Jellybean-based AVN systems, the navigation app data of the KitKat-based AVN system is created by HKMC_Navi_ENC.apk and stored in database (.db) or binary (.dat, .bin) formats under directory `/data/data/com.mnsoft.navi/`. The `/data/data/com.mnsoft.navi/databases/Navi_vr.db` file exists in both Jellybean-based and KitKat-based AVN systems. Among the navigation app data stored in the database, key artefacts include recent destinations, the recent search history, favourite locations, registered locations, etc.

By analysing the tables in the `Navi_vr.db` database, we found that two versions of the AVN system manage the five tables in common: `Current_Loc_Table`, `Destination_Table`, `NonSearch_Table`, `Memory_point_Table` and `RegisPnt_Special_Table`. Note that the name of the `Memory_point_Table` (in the Jellybean-based AVN systems) is slightly changed to `MemoryPoint_Table` in the KitKat-based AVN systems. Additionally, the attribute structure of the two tables (`MemoryPoint_Table` and `RegisPnt_Special_Table`) is different between the Jellybean and KitKat versions. Moreover, the KitKat-based AVN systems introduce new tables such as `RegisPnt_G1_Table`, `RegisPnt_G2_Table`, etc. Table 8 lists the tables and their attributes of `Navi_vr.db` in the KitKat-based AVN systems.

Table 8. Navigation data contained in the database file in the KitKat-based AVN systems.

File Name	Table	Attribute	Description
database/ Navi_vr.db	MemoryPoint_Table	Info_Name	My home
			My office
		Info_Address	location address
		Info_Longitude	longitude of the location
		Info_Latitude	latitude of the location
	RegisPnt_G#_Table (RegisPnt_G1_Table, RegisPnt_G2_Table, RegisPnt_G3_Table)	Info_Name	favourite location information
		Info_Address	favourite location's address
		Info_Tel	favourite location's phone number
		Info_Longitude	favourite location's longitude
		Info_Latitude	favourite location's latitude
	RegisPnt_Special_Table	Info_Name	registered location information
		Info_Address	registered location's address
		Info_Tel	registered location's phone number
		Info_Longitude	registered location's longitude
		Info_Latitude	registered location's latitude

The favourite locations are stored in `MemoryPoint_Table` along with the information of "My home" and "My office". The information of three places that the user can name arbitrarily are stored in `RegisPnt_G1_Table`, `RegisPnt_G2_Table`, and `RegisPnt_G3_Table`. The registered locations are stored in `RegisPnt_Special_Table`, and the basic information of the registered locations can be check through the attributes of this table.

There is also other navigation data that is not stored in these databases, such as destination search history, GPS records, special registered locations (or points-of-interest), user profile information, and recent search history. These data are in the `/oem_data/mnsoft/UserData/KOR/` directory, as shown in Table 9. Unlike the Jellybean-based AVN systems, a new directory USERPROFLE exists. The KitKat-based AVN system can set a user profile when a vehicle is used by multiple drivers. Thus, a driver can use the navigation system after selecting the corresponding profile. To reflect this upgraded function, the `USERPROFLE` directory stores the personalized information that each driver entered into the navigation module.

The personalized information is contained in a JavaScript Object Notation (JSON) file found in the `USERPROFLE` directory. The `arr_favorites` object of the `NavigationPOIAdd_Driver#.json` file keeps the favourite destinations, and home and work addresses set by a driver. The addresses can also be found in the `MemoryPoint_Table` and `RegisPnt_G#_Table` of `Navi_vr.db` file in Table 8. The JSON file does not exist in Jellybean-based AVN systems.

4.2.3. System Logs

System log data is located under the `/log` directory and most are stored in a text format. Communication data are stored in an encrypted format. Thus, we cannot analyse the Bluetooth communication logs. Through further inspection, we found that the `BluetoothContacts.db`, `MC_[MAC address].db` and `MF_[MAC address].db` files listed in Table 7 exist under the `/log/backup_data/` directory. However, we cannot find any time information in these databases.

Table 9. Navigation data in non-database files in the KitKat-based AVN systems.

Location	Directory/File Name	Description
UserData/KOR/	FavoriteDest.bin	search destination information
		search destination address
		search destination longitude/latitude
		search destination Korea area Code
		search time
	GPSTrack.dat	GPS record (longitude ∣ latitude ∣ Korea area Code)
	USERPOI	registered location information
		registered location address
		registered location longitude/latitude
	USERRECENT	Recently searched word
		The address a user has last searched for
		Recently searched destination's phone number
		Recently searched destination's longitude/latitude
		Recently searched destination's Korea area code
	USERPROFLE	favourite location info
		home address
		office address (work address)

In the system logs of the KitKat-based AVN systems, there were many logs of vehicle events that could not be identified in the Jellybean-based AVN systems. Table 10 shows the logs created by the `EngineIdleAlarmTask` process between 12:50 and 12:51 on 20 May 2020. The first row indicates that the vehicle's transmission was not in "Parking" and the vehicle was not in motion. The second row indicates that the vehicle's transmission was in "Parking" and the vehicle was not moving. The third row indicates that the vehicle's transmission was not in "Parking" and the vehicle was in motion. Therefore, it can be inferred that the driver set the transmission to "Parking" at 12:51:18, released the "Parking" status of the transmission at 12:51:47 and drove the vehicle.

Table 10. System log example under the `/log` directory.

Time	Process	Info
200520, 12:50:57.735	EngineIdleAlarmTask	ADM ON: false Gear in Parking: false Vehicle Moving: false Pressed Key: false
200520, 12:51:18.666	EngineIdleAlarmTask	ADM ON: false Gear in Parking: true Vehicle Moving: false Pressed Key: false
200520 12:51:47.739	EngineIdleAlarmTask	ADM ON: false Gear in Parking: false Vehicle Moving: true Pressed Key: false

4.3. Timeline Construction

Bluetooth and navigation data have relatively detailed information about the driver's activities, but do not have all the temporal information for each event. On the other hand, the system logs contain temporal information for the recorded events, but only store major events, making it difficult to systematically infer the driver's activities with the system logs. Through these findings, we can see that an integrated analysis is required for effective forensic analysis rather than individual analysis of the data or logs. We were able to construct a timeline of the drivers' activities by integrating then analyzing the Bluetooth data, navigation data, and system logs.

Table 11 shows the constructed timeline for the Kia NIRO EV. The driver started the vehicle at epoch time 1551854738. The `epoch converter` tool tells us that the epoch time 1551854738 was on 6 March 2019, at 6:45:38 Korean time. The driver searched for "office" as the destination, and he set the "office" as the destination at 6:45:45 a.m. on 6 March 2019. These activities were found by analysing the `startlog_1551854738.txt`, `Navi_vr.db`, `binary file`, etc., which are navigation data.

Table 11. Timeline of the driver's events/activities constructed from forensic artefacts of the Kia NIRO EV.

Time	Event	Artefact (Location and Name)	Description
6/3/2019 06:45:38	Vehicle ON	Navigation App (/data/data/com.mnsoft.navi/UserData /KOR/startlog_[epochtime].txt)	Driver starts the engine
6/3/2019 06:45:45	Destination search	Navigation App (/data/data/com.mnsoft.navi/databases /Navi_vr.db) Select the one registered in the info_name field of Destination_table	Chooses "office" as destination. (Address)
		Navigation App (/data/data/com.mnsoft.navi/Userdata /KOR/FavoriteDest.bin)	Time to search the destination
6/3/2019 06:52:47 ~6/3/2019 06:53:15	A smartphone is connected to the AVN via Bluetooth	Bluetooth App (/data/data/com.android.providers. bluetooth/databases/BTSetup.db)	MAC address and name of the smartphone
		System Log (/ivilog/dropbox/trace_log.txt)	Bluetooth connection time
	Downloading Phonebook	Bluetooth App (/data/data/com.android.providers. bluetooth/databases/BTContacts.db)	phonebook (names of people, phone numbers, etc.)
	Downloading recent call history	Bluetooth App (/data/data/com.android.providers. bluetooth/databases/BTCallHistory.db)	recent call history (names, phone numbers, time of the call)
6/3/2019 07:39:18	Phone call	Bluetooth App (/data/data/com.android.providers. bluetooth/databases/BTCallHistory.db)	Call info (name, phone numbers, time of the call)
6/3/2019 18:40:54	Current location	Navigation App (/data/data/com.mnsoft.navi /UserData/KOR/GPStrack.dat)	GPS tracklogs (longitude, latitude, Korea area code)

At 6:52:47 a.m. on 6 March 2019, the driver connected their smartphone Galaxy Note 8 to the car's AVN system via Bluetooth. The Bluetooth MAC address of the smartphone is 94:8b:c1:88:ba:b3. These activities were found by integrating the Bluetooth data with the system logs (e.g., class bt-hci) and analysing the integrated information.

After a Bluetooth connection is established between the two devices, the AVN system sends a message requesting access to the phonebook and recent call history of the smartphone. If a user accepts the request, the phonebook and recent call history are sent to and stored in the AVN system. We found that this transmission process was completed at 6:53:15 a.m. on 6 March 2019 by analysing the class (bt-hci) related to Bluetooth communication in the system logs. By analysing `BTCallHistory.db`, we also identified the name of the caller or receiver, their phone number, and the call time when the driver made the phone call through the AVN system.

Finally, we determined the current location of the vehicle at that time by analysing the last offsets 45625103 (0F 2F B8 45), 13475891 (33 A0 CD 00), and 1105385599 (7F D8 E2

41) in the `GPSTrack.dat` file which stored GPS tracklogs. The offsets, in turn, represent the longitude, latitude, and Korea area codes. Using the `Talmap` tool, we found that the offsets represent the address of 500, Sorae-ro, Namdong-gu, Incheon, Republic of Korea.

5. Discussion

5.1. Jellybean vs. KitKat

We investigated the Jellybean-based and KitKat-based AVN systems. We found differences in the methods for collecting forensic artefacts. At the forensic data acquisition stage, we could logically acquire the forensic data from the Jellybean-based AVN systems using ADB. However, since we could not use the ADB interface in the KitKat-based AVN systems, a chip-off method was used to obtain the physical image of an eMMC chip. At the forensic artefact analysis stage, we used the same tools, such as `Autopsy`, `HxD`, `Notepad`, `Talmap` and `Epoch Converter`, to analyse the collected data from both AVN system versions. As a result of the analysis, we were able to discover each database/file location and contents for the Bluetooth data, navigation data, and system logs. Table 12 summarizes the main artefacts obtained from the Bluetooth data, navigation data and system logs. It also analyses the key different artefacts between the two versions of the AVN systems.

When comparing the Jellybean-based and KitKat-based AVN systems, most of the Bluetooth data and navigation data were very similar, but there were some differences in the data storage location, file names, and detailed fields. In the KitKat-based AVN systems, a function to set each driver's profile was added to the navigation module, and we could analyse the saved profile information for each driver.

In both versions of the AVN systems, the system logs included common artefacts such as Bluetooth connection time, GPS tracklogs and their time stamp. However, the system logs in the Jellybean-based AVN systems stored the message reception records of the connected smartphone in plain text, while the system logs in the KitKat-based AVN systems stored this contents in an encrypted form. In addition, unlike the Jellybean-based AVN systems, there were additional logs related to vehicle events in the KitKat-based AVN systems. We were able to identify vehicle door opening/closing information, gear shift state, driving state, and more.

Table 12. Comparative analysis of the main artefacts between the Jellybean-based and KitKat-based AVN systems.

	Bluetooth Data	Navigation Data	System Logs
Common artefacts	- MAC address of mobile device - Mobile device name - Phonebook - Recent call history	- Recent destinations - Search history for recent destinations - Search history for destinations - Favourite and registered locations on the navigation - GPS tracklogs (longitude, latitude, etc.)	- Bluetooth connection time - GPS tracklogs (longitude, latitude) - Time stamp of the tracklogs
Different artefacts	- Storage paths for databases and files - File names	- Names of database tables - (Jellybean-based) Engine start time - (KitKat-based) Driver profile	- (Jellybean-based) Mobile phone text messaging log in plain text - (KitKat-based) Event log, Encrypted communication data

5.2. Comparison of Forensic Studies on AVN Systems

This section compares the results of the forensic analysis with state-of-the-art in AVN system forensics. Recent AVN systems support mirroring the display of connected smartphones on an AVN system's display. Android Auto and Apple CarPlay are smartphone applications that provide such a connectivity. Shin et al. [3] conducted forensic studies on AVN systems with Android Auto and Apple CarPlay. They divided the environment into four forensic targets: wireless communication between the cloud and smartphone, wireless communication between the vehicle and smartphone, and AVN system and smartphone internal storage. Since our research focused on the forensics of storage devices, we compare our research results with the analysis of AVN system and smartphone internal storage of [3]. We also compare our results with [8] that used a commercial tool, iVe.

Table 13 shows the comparison results. We collected data from four AVN systems. Data were acquired using logical extraction or chip-off techniques depending on the Android OS version. The target AVN system in [3] is Belsee's Best Aftermarket Auto, whose operating system is Android. Thus, many common artefacts were collected in our work and [3]. Since we collected data from various sources, we collected and identified more artefacts than others. In particular, through integrated analysis, our work can construct a timeline of user activity.

Table 13. Forensic studies on AVN Systems.

	Our Work	**Whelan et al. [8]**	**Shin et al. [3]**
Target AVN systems	LG LAN5020KKJF, LG IA88431DELE, Hyundai MOVIS 96560L1070SS, LG 965601Y000MB2	Uconnect 8.4, Toyota Extension Box	Belsee Best Aftermarket Auto
OS / file system	Android / ext4	Not available	Android / ext4, F2FS
Data acquisition method	logical extraction, chip-off	logical extraction	chip-off
Tools	Autopsy, HxD, DB4S, Notepad, dd, etc	iVe	DB4S, HxD, FTK Imager, TSK-based tool
Artifacts	connected device list, MAC address of device, name of device, connection time, contacts, call history, call logs, startlog, tracklog, search location, registered locations, home address, office address, door open/close info, gear shift state, driving state	IMEI, phone version, Apple id, last sync, contacts, call logs, audio files, locations, addresses	connected device list, connection time, phone number, location when paired, communication logs, app list in use, MAC address of vehicle, name of vehicle, last used time, disconnection time, activation time, WiFi connection history, Google Assistance history
Integrated Analysis	timeline		

5.3. Limitations

Our approach has some limitations. First, the presented approach may not be applied to QNX- or Windows CE-based AVN systems. We focused on collecting and analysing Bluetooth data, navigation data and system logs from Android-based AVN systems. Second, we did not address the acquisition and analysis of other data such as OBD-II, telematics, WiFi connection, etc. We need to develop a method to forensically analyse these data. Third, we employed a physical data acquisition method to collect data from the KitKat-based AVN systems. The physical data acquisition method is intrusive and requires a lot of time and effort compared with the logical data acquisition [6]. This means that a logical data acquisition method is needed for KitKat-based AVN systems. Fourth, the KitKat-based AVN system log communication data was encrypted; thus, we could not analyse the communication data. Furthermore, we could not find temporal information from the Bluetooth and navigation data in the Android-based AVN systems.

6. Conclusions

In this paper, we performed a digital forensic investigation of Jellybean- and KitKat-based AVN systems. AVN systems are often connected to the driver’s smartphone via Bluetooth. There were differences between the Jellybean- and KitKat-based AVN systems in the way of obtaining forensic images or imaging a storage. We used a logical data acquisition method for the Jellybean-based AVN systems, and a physical data acquisition method, chip-off, for the KitKat-based AVN system. We then identified and analysed the Bluetooth data, navigation data, and system logs of the image files using several digital forensic tools. The key digital artefacts of the AVN systems were stored in the form of databases, binary files, and text files.

The four AVN systems have common artefacts; however, the location and details of some artefacts were different. Of the system logs in the KitKat-based AVN systems, communication data was encrypted, thus, we could not analyse the communication data. The Bluetooth and navigation data in the AVN systems included information such as the phonebook of the previously connected smartphone, phone call history, and recent destinations, but they did not contain temporal information related to the information. On the other hand, the system log contained major events and their occurrence time. For systematic forensics, it is necessary to integrate and investigate these data. We thus were able to construct a timeline of a driver’s activities by integrating and analysing the Bluetooth data, navigation data, and system logs.

In the future, we plan to study logical data extraction methods and a more effective forensic investigation for KitKat-based AVN systems. We also plan to collect and analyse various artefacts by generating various driver events such as playing music, turning on digital multimedia broadcasting (DMB), playing online streaming media, etc. We also plan to collect and analyse artefacts generated by OBD-II, telematics, WiFi, etc.

Author Contributions: Investigation, H.K., H.S., I.K. and W.J.; writing—original draft, H.K., S.-J.C., M.P. and S.H.; writing—review and editing, S.-J.C., M.P. and S.H.; supervision, S.-J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science and ICT (no. 2021R1A2C2012574), and by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MIST) (No.2022-0-01022, Development of Collection and Integrated Analysis Methods of Automotive Inter/Intra System Artifacts through Construction of Event-based experimental system).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing the manuscript; nor in the decision to publish the results.

References

1. Fleming, B. Smarter cars: Incredible infotainment, wireless device charging, satellite-based road taxes, and better EV batteries [Automotive Electronics]. *IEEE Veh. Technol. Mag.* **2013**, *8*, 5–13. [CrossRef]
2. Garzon, S.R. Intelligent in-car-infotainment systems: A contextual personalized approach. In Proceedings of the 2012 Eighth International Conference on Intelligent Environments, Guanajuato, Mexico, 26–29 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 315–318.
3. Shin, Y.; Kim, S.; Jo, W.; Shon, T. Digital Forensic Case Studies for In-Vehicle Infotainment Systems Using Android Auto and Apple CarPlay. *Sensors* **2022**, *22*, 7196. [CrossRef] [PubMed]
4. Kopencova, D.; Rak, R. Issues of vehicle digital forensics. In Proceedings of the 2020 XII International Science-Technical Conference on Automotive Safety, Kielce, Poland, 21–23 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.

5. Lacroix, J.; El-Khatib, K.; Akalu, R. Vehicular digital forensics: What does my vehicle know about me? In Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, Malta, Malta, 13–17 November 2016; pp. 59–66.
6. Bortles, W.; McDonough, S.; Smith, C.; Stogsdill, M. *An Introduction to the Forensic Acquisition of Passenger Vehicle Infotainment and Telematics Systems Data*; Technical Report; SAE Technical Paper; SAE International: Warrendale, PA, USA, 2017. Available online: https://www.sae.org/publications/technical-papers/content/2017-01-1437/ (accessed on 18 March 2023).
7. The iVe Ecosystem. Available online: https://berla.co/ecosystem/ (accessed on 18 March 2023).
8. Whelan, C.J.; Sammons, J.; McManus, B.; Fenger, T.W. Retrieval of infotainment system artifacts from vehicles using iVe. *J. Appl. Digit. Evid.* **2018**, *1*, 30.
9. Le-Khac, N.A.; Jacobs, D.; Nijhoff, J.; Bertens, K.; Choo, K.K.R. Smart vehicle forensics: Challenges and case study. *Future Gener. Comput. Syst.* **2020**, *109*, 500–510. [CrossRef]
10. Lacroix, J. Vehicular Infotainment Forensics: Collecting Data and Putting It into Perspective. Ph.D. Thesis, University of Ontario Institute of Technology, Oshawa, ON, Canada, 2017.
11. Seong, H.; Lee, K.; Cho, S.J.; Han, S.; Park, M. A Preliminary Forensics Analysis of Navigation Records on an Android-based Audio-Video Navigation System. In Proceedings of the ICNGC 2021 Conference, Jeju, Korea, 4–6 November 2021.
12. Jacobs, D.; Choo, K.K.R.; Kechadi, M.T.; Le-Khac, N.A. Volkswagen car entertainment system forensics. In Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia, 1–4 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 699–705.
13. Kim, B.; Park, S. ECU software updating scenario using OTA technology through mobile communication network. In Proceedings of the 2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS), Singapore, 28–30 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 67–72.
14. In-Vehicle Infotainment System—Everything You Need to Know About. Available online: https://www.einfochips.com/blog/everything-you-need-to-know-about-in-vehicle-infotainment-system/ (accessed on 18 March 2023).
15. Hyundai Motor Group—Infotainment (in Korean). Available online: https://www.hyundai.co.kr/tech/2052 (accessed on 18 March 2023).
16. Hyundai Global Service Way—Technical Information (in Korean). Available online: https://gsw.hyundai.com/ (accessed on 18 March 2023).
17. Kia Global Service Way—Technical Information (in Korean). Available online: https://gsw.kia.com/ (accessed on 18 March 2023).
18. Buquerin, K.K.G.; Corbett, C.; Hof, H.J. A generalized approach to automotive forensics. *Forensic Sci. Int. Digit. Investig.* **2021**, *36*, 301111. [CrossRef]
19. Scientific Working Group on Digital Evidence (SWGDE)—Best Practices for Vehicle Infotainment and Telematics Systems. Available online: https://www.irisinvestigations.com/wp-content/uploads/2019/05/SWGDE-Best-Practices-for-Vehicle-Infotainment-and-Telematics-Systems-062316.pdf (accessed on 18 March 2023).
20. Mansor, H.; Markantonakis, K.; Akram, R.N.; Mayes, K.; Gurulian, I. Log your car: The non-invasive vehicle forensics. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 974–982.
21. OpenText Encase Forensic. Available online: https://www.opentext.com/products/encase-forensic (accessed on 18 March 2023).
22. FTK® Forensic Toolkit—Exterro. Available online: https://www.exterro.com/forensic-toolkit (accessed on 18 April 2023).
23. Autopsy—Digital Forensics. Available online: https://www.autopsy.com/ (accessed on 18 April 2023).
24. X-Ways Forensics: Integrated Computer Forensics Software. Available online: https://www.x-ways.net/forensics/ (accessed on 18 April 2023).
25. Andriller—A Collection of Forensic Tools for Smartphones. Available online: https://github.com/den4uk/andriller (accessed on 22 April 2023).
26. Marturana, F.; Me, G.; Tacconi, S. A case study on digital forensics in the cloud. In Proceedings of the 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Sanya, China, 10–12 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 111–116.
27. Zhang, X.; Upton, O.; Beebe, N.L.; Choo, K.K.R. Iot botnet forensics: A comprehensive digital forensic case study on mirai botnet servers. *Forensic Sci. Int. Digit. Investig.* **2020**, *32*, 300926. [CrossRef]
28. Salamh, F.E.; Mirza, M.M.; Karabiyik, U. UAV forensic analysis and software tools assessment: DJI Phantom 4 and Matrice 210 as case studies. *Electronics* **2021**, *10*, 733. [CrossRef]
29. LG Electronics—LG Open Source. Available online: http://opensource.lge.com/product/list?ctgr=024&subCtgr=051 (accessed on 18 March 2023).
30. Kia Corp.—Official Kia Navigation Update Website. Available online: https://update.kia.com/US/EN/navigationUpdate (accessed on 18 March 2023).
31. Kernel_exploitation/CVE-2016-5195. Available online: https://github.com/N1rv0us/kernel_exploitation/tree/55a2aff8b6620bf8a59612bcc5796a0bcbbfdf71/CVE-2016-5195/poc (accessed on 18 March 2023).
32. Liu, L.S.; Lin, J.F.; Yao, J.X.; He, D.W.; Zheng, J.S.; Huang, J.; Shi, P. Path planning for smart car based on Dijkstra algorithm and dynamic window approach. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8881684. [CrossRef]
33. Pasandi, L.; Hooshmand, M.; Rahbar, M. Modified A* Algorithm integrated with ant colony optimization for multi-objective route-finding; case study: Yazd. *Appl. Soft Comput.* **2021**, *113*, 107877. [CrossRef]

34. Chen, Y.; Cheng, C.; Zhang, Y.; Li, X.; Sun, L. A neural network-based navigation approach for autonomous mobile robot systems. *Appl. Sci.* **2022**, *12*, 7796. [CrossRef]
35. TalMap. Available online: http://www.talmap.co.kr/ (accessed on 18 March 2023).
36. Yang, J.O.; Bang, M.J.; Lee, S.W.; Cho, T. Identification of the Crime Scene through Bluetooth HCI Snoop Log (in Korean). In Proceedings of the Korea Information Processing Society Conference. Korea Information Processing Society, Busan, Korea, 2–3 November 2018; pp. 249–252.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.