

**VEHICLE AND MOBILE APPLICATIONS INTERACTION ANALYSIS:
DIGITAL FORENSICS APPROACH**

by
Qiyuan Li

A Thesis

*Submitted to the Faculty of Purdue University
In Partial Fulfillment of the Requirements for the degree of*

Master of Science



Department of Computer and Information Technology
West Lafayette, Indiana
May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Umit Karabiyik, Chair

Department of Computer and Information Technology

Dr. Baijian Yang

Department of Computer and Information Technology

Dr. Jin Wei-Kocsis

Department of Computer and Information Technology

Approved by:

Dr. Dr. John A. Springer

With this opportunity, I would also like to thank all of my family members who helped me through this research process at this special time.

ACKNOWLEDGMENTS

I want to send my sincere appreciation to my advisor professor, Dr. Umit Karabiyik, for all the inspirational opinions and guidance. Additionally, I would like to thank Dr. Baijian Yang and Dr. Jin Wei-Kocsis, my committee members, for their guidance in my research work.

I would also like to give special thanks to my partner, and all my friends. It would be impossible for me to complete my study, without the support and understanding from all of them during this difficult time.

Hope all of us and our loved ones stay strong, safe, and healthy.

TABLE OF CONTENTS

LIST OF TABLES	7
LIST OF FIGURES	8
ABSTRACT	10
CHAPTER 1. INTRODUCTION	11
1.1 Background	11
1.2 Research Justification	12
1.3 Research Questions	13
1.4 Assumptions.....	13
1.5 Limitations	14
1.6 Delimitations.....	14
CHAPTER 2. LITERATURE REVIEWS	15
2.1 Vehicle Forensics.....	15
2.2 Mobile and Internet of Things (IoT) Forensics.....	18
CHAPTER 3. METHODOLOGY.....	22
3.1 General Framework	22
3.2 Test Devices.....	24
3.3 Forensic Tool and Working Environment	26
3.4 Data Population.....	27
3.4.1 Testing Route.....	27
3.4.2 Test Cases	28
3.5 Data Extraction	29
CHAPTER 4. Results.....	30
4.1 iOS Device analyzing	30
4.1.1 Nonda ZUS - iOS.....	30
4.1.2 Mercedes-Benz Companion iOS	42
4.2 Android Device Analyzing	47
4.2.1 Nonda ZUS Android.....	47
4.2.2 FIXD Android.....	63
CHAPTER 5. DISCUSSION AND FUTURE WORK.....	65

5.1	Research Question	65
5.1.1	Research Question 1	65
5.1.2	Research Question 2	65
5.1.3	Research Question 3	66
5.1.4	Research Question 4	67
5.2	Recovered Artifacts Review	68
5.2.1	Mercedes Benz Companion – iOS.....	68
5.2.2	Nonda ZUS – iOS.....	68
5.2.3	Nonda ZUS – Android.....	69
5.2.4	FIXD – Android.....	69
5.3	Suggestions and Recommendations to the Digital Forensic Investigators	70
5.4	Limitations During the Research	70
5.4.1	Data Population	70
5.4.2	Data Extraction	71
5.4.3	Data Analysis.....	71
5.5	Future Work.....	72
REFERENCES		73

LIST OF TABLES

Table 3.1 Test Mobile Devices	24
Table 3.2 Test Vehicles.....	25
Table 3.3 Test Mobile Application	25
Table 3.4 Information for the Forensics Tools	26
Table 3.5 Workstation Configuration Information	27
Table 3.6 All Test Cases Information	29
Table 3.7 Detailed Information for Forensics Images	29
Table 5.1 Comparison of geolocation related artifacts found from both iOS and Android devices	65
Table 5.2 Comparison of vehicle related artifacts found from all Four testing Applications	66
Table 5.3 Comparison of vehicle-related artifacts found from both devices for different vehicles	66
Table 5.4 Comparison of artifacts found from the different mobile devices.....	67

LIST OF FIGURES

Figure 3.1 Diagram of the Research Framework: Third-party Applications.....	23
Figure 3.2 Diagram of the Research Framework: Native Application	24
Figure 3.3 Road Map for the Testing Rout	28
Figure 4.1 Username and email Address Part 1	31
Figure 4.2 Username and email Address Part 2.....	31
Figure 4.3 User Vehicle Count	31
Figure 4.4User Vehicle ID Number Part 1	32
Figure 4.5User Vehicle ID Number Part 2	32
Figure 4.6OBD II Adaptor Model Information for Vehicle 1	33
Figure 4.7OBD II Adaptor Model Information for Vehicle 2	33
Figure 4.8 Vehicle DTC Scanning Log	34
Figure 4.9 File Information for Trip snapshot 0a533abe923b945f870029a9292d6989	35
Figure 4.10 Trip snapshot 0a533abe923b945f870029a9292d6989.....	35
Figure 4.11 Trip snapshot 61b84eb8150fc1354f5a82057d32d350	36
Figure 4.12 Trip snapshot 04914c36303dc1547b5c9ae2e08e3247	36
Figure 4.13 Trip snapshot b71577a3c670c14b403e60a37ab653b5.....	37
Figure 4.14 Trip snapshot bea57c38f837f495626d055711e6bfc2	37
Figure 4.15 Trip snapshot db6143538dc868e47060685570a32cf1	38
Figure 4.16 Motion Activity Log and Translation.....	39
Figure 4.17 Geofence Information in the Application Log File	39
Figure 4.18 Geofence Information Captured by the Cellebrite UFED Reader.....	40
Figure 4.19 Parking Information Found in the Application Log file.....	41
Figure 4.20 Parking Finder Count	42
Figure 4.21 File System for MB-Companion Application Package	43
Figure 4.22 Error Message from SQLiteSPY	44
Figure 4.23 Error Message from DB Browser for SQLite.....	44
Figure 4.24 Records Found from data.data File	45

Figure 4.25 Metadata for KTX Image Files.....	46
Figure 4.26 Converted KTX Image Files	46
Figure 4.27 User Account Information from XML File	48
Figure 4.28 User Account Information from realm Database	48
Figure 4.29 User Vehicle ID Information (with VIN).....	49
Figure 4.30 User Vehicle ID Information (without VIN) Part 1	49
Figure 4.31 User Vehicle ID Information (without VIN) Part 2	50
Figure 4.32 User Vehicle Detailed Information	51
Figure 4.33 User Vehicle Configurations Information	51
Figure 4.34 OBD II Adaptor ID Number from Log Files.....	53
Figure 4.35 OBD II Adaptor Model from Log Files	53
Figure 4.36 OBD II Adaptor Detailed Information from the realm Database.....	53
Figure 4.37 Four Records of the Result from Vehicle Safety Scanning.....	54
Figure 4.38 Two DTC Records with Detailed Info	55
Figure 4.39 File Information for Trip Snapshot 4c1704a21b94871c1d1661b0137b4e6dd3bd296f1cb96d514129d7d39103930b.0	56
Figure 4.40 Trip Snapshot 4c1704a21b94871c1d1661b0137b4e6dd3bd296f1cb96d514129d7d39103930b.0	56
Figure 4.41 Detailed trip Information from realm database part 1	57
Figure 4.42 Detailed trip Information from realm database part 2	58
Figure 4.43 Detailed trip Information from realm database part 3	59
Figure 4.44 Detailed GPS Information from the realm database.....	60
Figure 4.45 Trip snapshot for trip ID 3f38e4694b6d411988497a183bd10ae1	61
Figure 4.46 Recreated route map for trip ID 3f38e4694b6d411988497a183bd10ae1	61
Figure 4.47 Geofence Information from the realm database	62
Figure 4.48 Parking Information from the realm database	63
Figure 4.49 User email address from the XML file.....	63
Figure 4.50 Vehicle's VIN Information from the XML file.....	64

ABSTRACT

With the Internet of Things (IoT) development, vehicles have become an essential part of this data transmission network. In order to access the vehicle's status via personal mobile devices, an increasing number of car manufacturers have begun to provide mobile applications; some third-party companies offer Bluetooth adaptors for the On-Board Diagnostics-II (OBD-II) port on vehicles made post-1996 in the United States. By connecting the smartphone and the vehicle with either of these methods, the mobile applications can retrieve detailed data and the history of the vehicle. This research aims to answer what forensically relevant artifacts can be recovered from the MB Companion, FIXD, and Nonda ZUS applications. The research methods include adapting the National Institute of Standards and Technology (NIST) forensics framework, generating mock user data, extracting user data, and conducting in-depth digital forensics analysis. The recovered geolocation data, the vehicle-related artifacts, the applications on different vehicle brands, and the applications on various device platforms are primarily examined in the research.

CHAPTER 1. INTRODUCTION

1.1 Background

Nowadays, most of the devices in modern life are connected to the Internet of Things, including personal vehicles. According to the Bureau of Transportation Statistics (2017), eighty-seven percent of daily trips were made via personal vehicles, and ninety-one percent of people in the United States commuted to work by driving personal vehicles. The number gives us a basic idea of how likely a personal vehicle can get involved in a criminal case, which is remarkably high. The study of the brain of the modern vehicles—ECU (Electronic Control Unit)—will be essential due to the amount of data it stores.

In the past, communicating with the ECU was a privilege held only by car manufacturers, dealers, and auto repair shops. If car owners want to know what is wrong with their vehicles, they have to pay extra money to the mechanic to check it out. However, this situation is changed now. An increasing number of car manufacturers are making matched mobile applications for the owners. For example, Mercedes Benz (MB) provides an application called "Companion" for Mercedes vehicles built after 2015. This mobile application allows the owners to connect their smartphone with the vehicle via Bluetooth. Once connected, the user can view their vehicle's status and send the destination to the infotainment system. The mobile application can also record the parking location and trip history, which is crucial as evidence for digital forensics investigation.

However, what about the older models that still run on the road? There are products for any vehicles made post-1996 in the United States equipped with the OBD-II (On-Board Diagnostics Second Gen) port, which was initially designed for 16-pin connectors used by authorized personnel like mechanics. Two of their products are called FIXD and Nonda Zus, which are OBD-II Bluetooth adaptors. When the products are plugged into the port and connected with

a smartphone with the matched application installed, older vehicles made after 1996 could also become a part of the IoT. These paired mobile applications allow complete scanning of the vehicle, so the user can find fault codes and fix them before the problems become severe. Moreover, such applications even provide functions like recording the trip history and parking location.

1.2 Research Justification

In 2017, 87% of people living in the United States drove their vehicles around every day. According to Meola (2020), thirty-three million smart vehicles were running on the road, and they were a part of the IoT in 2017, and this number will reach seventy-seven million by the year 2025. With this amount of intelligent vehicles around, the possibility that any of them being involved in a criminal case would definitely get higher.

On the other hand, even for cars that are unable to connect with the IoT network, the appearance of products like Nonda and FIXD makes this communication channel open. According to the FIXD's market promotion (*FIXD Official Site*, 2021), they have sold more than two million OBD adaptors in the United States, and there have been more than five hundred thousand installations just on the Google Play store (*FIXD - Vehicle Health Monitor - Apps on Google Play*, 2022).

Currently, the established vehicle-related digital forensic research focuses on hardware and unutilized framework. For instance, there are studies about the hardware on vehicle's infotainment systems (Le-Khac et al., 2020) and designing a framework that allows the vehicles to share data with specific stakeholders, like car manufacturers or law enforcement (Cebe et al., 2018). However, the research on the native and third-party vehicle mobile device connection applications has not been completely covered. This research would fill the gap in vehicle forensics by studying the file

system and the data chain on both Android and iOS with native and third-party vehicle interactive applications.

1.3 Research Questions

The main goal of this research is to answer the following question:

- What forensically relevant artifacts can be recovered from the MB Companion, FIXD, and Nonda ZUS applications?

The following sub-questions will be focused on during this research:

1. Will complete route information be pictured by the geolocation data recovered from MB Companion and Nonda ZUS?
2. What different vehicle information can be recovered from the MB Companion, FIXD, and Nonda ZUS applications?
3. What different vehicle information can be recovered from the Mercedes and the Audi?
4. What impacts from using different devices (Android and iOS) will occur when recovering the relevant artifacts?

1.4 Assumptions

This research has the following assumptions:

- The iOS and Android phones have not been modified for the hardware.
- The objective applications are installed and used by the user.
- The data are not altered after jailbreaking and rooting the testing devices.

- After the data generating stage, the data are not altered before making the forensics image.
- The researcher has access to smartphones.

1.5 Limitations

This research has the following limitations:

- Many first-party car manufacturers offer mobile applications connecting to their vehicles, like Hyundai, Tesla, and Mercedes. This research only focuses on the Mercedes OEM (Original equipment manufacturer) mobile application.
- Many third-party parts manufacturers offer OBD-II Bluetooth adaptors. However, this study only focuses on FIXD and Nonda adapters and the matched mobile applications.
- Users are able to perform multiple activities with the mobile applications, such as scanning the vehicle and generating reports, marking the parking location on the app, setting the destination on the application, and sending it to the vehicle. However, not all functions were tested in this study.

1.6 Delimitations

This research has the following delimitations:

- Only one iOS and one Android device will be studied in this study.
- Only three applications (Mercedes Companion, Nonda ZUS, and FIXD) will be studied in this study.
- No hardware forensics will be performed during the study.

CHAPTER 2. LITERATURE REVIEWS

Vehicles involved in criminal scenes are used to glean physical forensic evidence such as fingerprints, DNA, and other non-digital materials. With the rapid development of information technology, digital forensics has progressed in the meantime. Digital forensics is defined as “the application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody of the data” (Kent et al., 2006, p. 2-1). With the advances in technology, more and more vehicles' embedded systems become the key source of digital evidence in criminal investigations as they can give access to digital evidence (Alexakos et al., 2021; Kopencova & Rak, 2020; Le-Khac et al., 2020; Mylonas et al., 2012). They can store a vast amount of digital information such as favorite locations and destinations, recent destinations, and routes. Additionally, modern-day cars can store personal data such as contact lists, call logs, short messages, videos, and pictures. Analysis of current and related research from fields such as android mobile, car rental, smart cities, cybersecurity, and how these topics relate to vehicle forensics is necessary.

2.1 Vehicle Forensics

One of the modern vehicles' main components is the infotainment system, which allows the vehicle to generate, log, and exchange data about the vehicle and its surroundings. Devices such as smartphones and laptops can send information to the infotainment system through Bluetooth and vehicle Wi-Fi (Bortles et al., 2017). As a result, trends, activities, and the localized data embedded in the GPS (Global Positioning System) are installed in the infotainment system. Mining such information would help the Intelligent Transport system and Vehicular Ad Hoc Network in investigating crimes.

Modern vehicles are also embedded with other digital devices, such as event data recorders (Gabler et al., 2008). Lacroix et al. (2016) analyzed and validated the testing results of Event Data Recorders (EDR) and found out that they were accurate and underreported the vehicles' pre-crash speed. As such, these devices can be used to reconstruct the accident and provide valid data that can be used in digital forensics investigations. Additionally, data from GPS devices can provide insightful information regarding the vehicle and the driver. Such data can be used to investigate an accident involving a car. Some of the critical applications in the GPS that can be used to supplement criminal investigations include vehicle navigation and automatic vehicle location (Chang & Cheon, 2019; Morgul et al., 2013; Zunic et al., 2018). Similar applications contain fleet management networks, autonomous vehicle operation, and automatic collision notifications (Campbell et al., 2018; Malekian et al., 2017; White et al., 2011). Therefore, harvesting and analyzing such information would improve vehicle safety, as well as harness criminal investigations.

Another real-world example of applying IoT devices for car rental management is ForRent. The car rental system is plagued by rental damage frauds where the renter fails to produce evidence against being charged for damages that did not happen during his rental. FoRent is a mobile application diagnostic feature where the car renter and the owner can record and retrieve the car's physical appearance through the ELM327 device (Saufi et al., 2019). ELM327 is a programmed microcontroller that decodes the on-board diagnostics (OBD) and recovers forensic data from the Electronic Control Unit (ECU). The data is synchronized and then displayed when connected to an application through Bluetooth. Hence, FoRent justifies the two parties as it can provide valuable insights from forensic data records, fault code systems, and accidents during the rental period. Similarly, it can be used as an accessory for forensic analysts.

A case study composed by Dr. Le-Khac's research team (2017) discussed a suspect who drove a rental 2012 Volkswagen Golf and returned this vehicle to the rental company. A research group of forensics researchers and police investigators was working to see if they could extract any useful data from the car's entertainment system that could be used as evidence against the suspect in the court. In this case, the research team found some data from this 2012 Volkswagen Golf's RNS-510 infotainment system by using the Joint Test Action Group (JTAG) on the on-board Random Access Memory (RAM) (Neeb, 2004). Also, they chip-offed four memory chips (Fukami et al., 2017). However, most of the recovered artifacts they found from the infotainment system were not particularly useful for the case. Such artifacts include the map Information for the built-in navigation system and the mp3 playlist. But the most important thing was that all those data were forensically sound. This research clarified that the car's entertainment system stores data all the time. However, the small amount of capacity of those on-board RAMs and built-in Hard Disk Drive (HDD) makes the system reset the storage frequently. Even though the designated functionality forces the system to keep whipping the existing data, when the car is connected to a mobile device like a smartphone via specific applications or third-party OBD readers, the smartphone could store those vehicle data on itself or upload them to the cloud. It gives the investigators the possibility to extract those data as evidence.

A research team from the University of London (Mansor et al., 2016) found the EDR and black box on vehicles only contain elementary functions. For example, if the testing vehicle has some electrical issues, the EDR will stop recording, which happens a lot for most vehicles running outside. To solve this problem, they introduced a mobile application called DiaLOG. This app connects to the vehicle, receives the Diagnostic Transmission Codes (DTCs), and stores them

safely on the mobile application and the cloud. This framework allows the forensics investigators to work more effectively and maintain the user's data simultaneously.

Last but not least, a team from Florida International University (Cebe et al., 2018) designed a framework based on a permissioned blockchain. This framework allows the vehicle-related data gleaned from hundreds of sensors built in the vehicle to be decentralized and stored in different nodes when the vehicle has been crashed, lost, or in other situations where the investigators cannot access the physical storage devices. Cebe et al. (2018) introduced a framework called Block4Forencis, which connects the vehicle, service provider, car manufacturer, law enforcement, and insurance company. They planted a forensics daemon in the On-Board Unit (OBU) on the object vehicle so that this daemon could retrieve data from the EDR, Basic Safety Message (BSM), and all the sensors on the vehicle. The hashed data will send through the blockchain, and only certified stockholders will be able to receive and decrypt these data. This could help the scene's reconstruction to be much more comfortable and creditable. The law enforcement investigators could find more integrity and encrypted data if the vehicle continued to get involved in any criminal cases.

2.2 Mobile and Internet of Things (IoT) Forensics

Another topic related to vehicle forensics is Smart City (Feng et al., 2017., 2019; Losavio et al., 2018; Tang et al., 2020). There is a network of digital devices that are constantly exchanging data to improve living standards. The cities usually involve systems such as smart grids, building automation systems, autonomous aerial vehicles, as well as intelligent cars. These cities will be made of digital sensors that facilitate smart parking, optimize routes, monitor traffic, and enable street lighting. The data collected from the smart grid devices can be used to study consumption patterns and demand and supply management. Additionally, cloud computing will foster the

storage and processing of large volumes of data that can be used to detect strange behaviors in smart grids, which is essential for forensic investigations.

Smartphones are also critical IoT devices that can be valuable in vehicle forensics as they merge with a person's everyday life (Ebberts et al., 2021; Prastya et al., 2017; Saufi et al., 2019). Further, they are mobile and can store a wide variety of data used for forensic investigations. For instance, Al-Sabaawi and Foo (2019) indicate that these devices' data can be proactively acquired to investigate crimes against the public or the state. Through lawful interception, data such as phone calls, messages, and network traffic data can be accessed to supplement investigations. By analyzing two popular third-party location applications, Bays and Karabiyik (2019) found out that both the user and the user's contacts' GPS information can be investigated through digital forensics, which facilitates addressing crimes including kidnapping and missing persons. Traditional forensic methods such as postmodern required forensic experts to acquire a forensic triage before accessing a crime scene. This often resulted in delays due to the long lab process. However, digital forensics using smartphones contains a wide variety of evidence such as identity, location, time, context, and motivation evidence as to how an event took place or the means used.

The high technological systems installed in modern cars and integration with the Internet and the satellite navigation system have increased car theft rates. As a result, it is essential to protect vehicles by increasing car security. Car security can be enhanced by using IoT devices that send a security threat message to the car owner in advance, help car owners get rid of bad and unintentional events, and provide access to the vehicle through the Internet when it is stolen or damaged (Mukhopadhyay et al., 2018; Ruengittinun et al., 2017; Sehgal et al., 2016). Such innovations would protect against threats, hacking, as well as accidents. Modern vehicles are also

installed with a black box that logs all occurrences. Such data can be retrieved in case of an accident and examined by forensic experts to determine what happened.

IoT devices can be used in many settings such as homes, society, battlefields, and national security (Choo et al., 2021; Jayakumar et al., 2016; Lott et al., 2019; Meneghello et al., 2019; Stoyanova et al., 2020; Suri et al., 2016; Wurm et al., 2016). For example, the Amazon echo in home settings can record data such as voice, weather patterns, and other real-time information (Li et al., 2019). These devices are mainly targeted by attackers (Jiang et al., 2020; Shah & Sengupta, 2020; Sikder et al., 2018; Sun et al., 2019), making them a key source of digital forensic investigations as incriminating data may be present in these devices.

Given the rapid evolution of IoT, digital forensics is becoming a challenging endeavor (Caviglione et al., 2017; Montasari & Hill, 2019). Salamh et al. (2021) emphasize that digital forensics must deal with continuously changing challenges in the context of the rapid development of software and hardware and identify privacy and security concerns in current applications. Further, cybercrime is projected to increase due to the many digital devices (Holt et al., 2017), the relevance of the collected devices, and edgeless networks that can be used to provide evidence (Baig et al., 2017). Such data may come from external hard drives, laptops, USB drives, as well as mobile devices. As a result, there will be a mass amount of data to analyze, which may be time-consuming, especially if there is no clear objective in the investigation (MacDermott et al., 2018). Similarly, there is a need to develop specialized tools to retrieve information from digital devices and new methods for storing digital evidence. Hence, it is essential to establish standards to deal with digital evidence, as such data can be used as an accessory to supplement forensic investigations.

The Internet of things has enabled vehicle forensics to be connected to other fields such as cybersecurity and smart cities, which has increased the amount of digital data as well as evidence that can be used in forensic investigations. Such data include:

- The driver's favorite routes
- Personal information
- Data logged in the black box can be used to reconstruct events

However, these digital devices have increased the difficulty in forensic investigation as the vast amount of data can be challenging to analyze objectively. Thus, there is a need to develop new standards for novel tools and standards for dealing with digital data in forensic investigations.

CHAPTER 3. METHODOLOGY

The research and studies that were discussed in the previous chapter mainly focused on hardware forensics and designing a framework or system that allows the smartphone to be a receiver for the vehicle, and they are inspirational. However, little to no research has focused on any established native or third-party car connecting applications. The research and studying objects in this thesis will locate the gap and fill it eventually. This chapter will clarify the research framework, research environment, study objects, and data population methods.

3.1 General Framework

The forensics framework adopted in this research is based on the guide introduced by the NIST (Kent et al., 2006), which provided multiple standards and helped in locating the scope of the research. This experimental study has a general framework that includes the following nine stages:

- Factory reset the test devices
- Creating Google Play and Apple ID accounts
- Installing test application
- Data population
- Jailbreaking and rooting the test devices
- Data extraction/acquisition
- Evidence collection

- Evidence verification
- Result organization

There were two types of testing devices applied during the study. The first is the vehicles, and the second part is the smartphones that have the application installed. The Android and iOS mobile devices were connected to the vehicles via the build-in OEM Bluetooth channel or the aftermarket OBD-II adapter made by Nonda and FIXD. The researcher drove the testing vehicles around when connected to the mobile devices during the data population. A testing log will be made for tracking the activities that have been created. Advanced logical images were made with the test phones via the famous mobile forensics tool, Cellebrite UFED Physical Analyzer 7.44. Such a tool is commonly used in the digital forensics field.

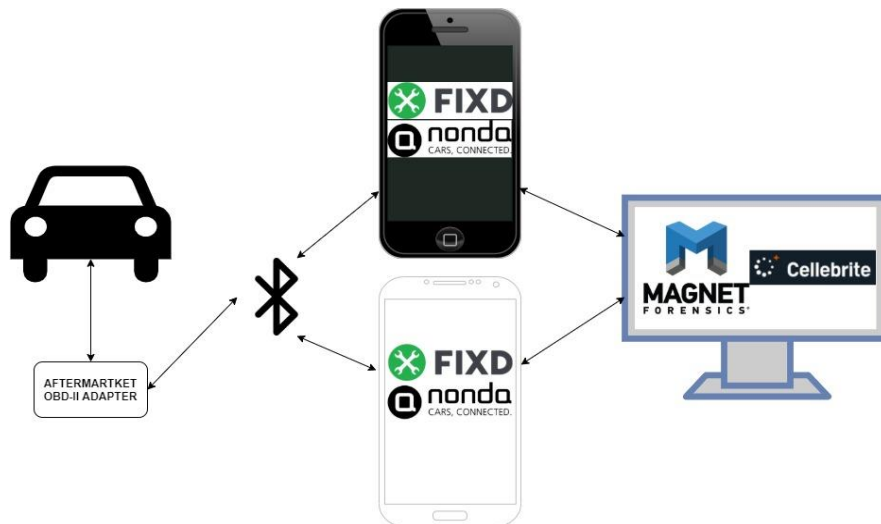


Figure 3.1 Diagram of the Research Framework: Third-party Applications

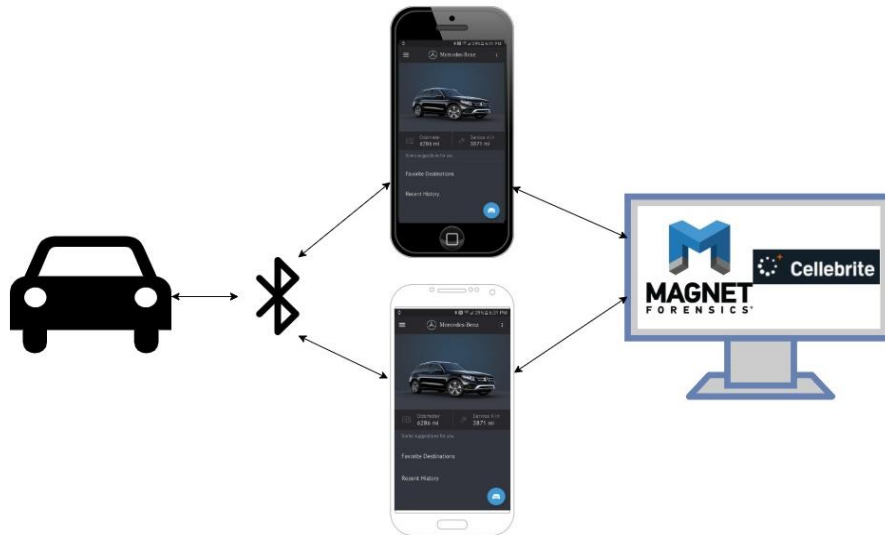


Figure 3.2 Diagram of the Research Framework: Native Application

3.2 Test Devices

There are three parts of testing devices in this study: the testing vehicles and the testing smartphones. Table 3.1 lists the information on the mobile test devices.

Table 3.1 Test Mobile Devices

Device Specification	Device #1	Device #2
Device Model	Galaxy S6	iPhone X
Manufacturer	SAMSUNG	Apple
Operating System	Android 7.0	iOS 14.3
IMEI	352885870137914	35485709176697

The second part is testing vehicles. Two testing vehicles were used to generate data for the research. The following table 3.2 lists the information of the testing vehicles.

Table 3.2 Test Vehicles

Vehicle Specification	Vehicle #1	Vehicle #2
Vehicle Model	A3	C-Class
Manufacturer	Audi	Mercedes
Built Year	2015	2016
Connection Channel (OEM / Third-party)	OBD-2 Third-party adapter	OEM Bluetooth and OBD-2 adapter

The third part is testing mobile applications. The following Table 3.3 lists the information on the applications that were studied in the research.

Table 3.3 Test Mobile Application

Application Data	Application #1 (Native app)	Application #2 (Third party)	Application #3 (Third party)
Application Name	MB Companion	ZUS	FIXD
Mobile App Version	iOS: 1.3.6	Android: 7.3.0 iOS: 7.3.0	Android: 7.18.1
Company Name	MB R&D NA	No NDA	FIXD Automotive
Operating System	iOS	Android & iOS	Android
OEM or Third-party	OEM	Third-party	Third-party

3.3 Forensic Tool and Working Environment

Two well-known forensic tools were applied in this research. The first forensic tool is called Cellebrite UFED 4PC, and it was used to generate advanced logical images. The second tool is called Cellebrite UFED Physical Analyzer, which is used as an examination tool to analyze the acquired mobile device images. Table 3.4 shows the detailed information for the forensic tools in this study.

Table 3.4 Information for the Forensics Tools		
Application Info	Acquisition Application	Examination Application
Name	UFED 4PC	UFED Physical Analyzer
Company	Cellebrite	Cellebrite
Release Version	7.44	7.42.0.50
Available OS	Windows	Windows
Physical or Logical	Advanced Logical	Advanced Logical

Those forensics tools were installed on a Windows 10 64-bit workstation. The reason for choosing a Windows workstation is that both forensics tools are available for Windows 10 operating system. Table 3.5 shows the operating system and hardware configuration for the workstation.

Table 3.5 Workstation Configuration Information

Workstation Configuration	Detailed Information
Name	Dell OptiPlex 7060 MFF
CPU	Intel Core i7-8700 @ 3.20 GHz
Operating System	Windows 10 Education
OS Version	1809

3.4 Data Population

The researcher generated the data, which was soon extracted and analyzed during the study. After successfully installing the mobile application into the mobile test devices, the connection between the testing vehicle and the mobile devices was established. Then, the researcher drove the testing vehicle following the designated route with testing mobile devices connected. Different test cases were designed and conducted. For example, Vehicle #1 followed route #1 and connected mobile device #2 via the OEM Bluetooth application.

3.4.1 Testing Route

The testing field of the study was mainly on the open road in West Lafayette, Indiana. A designated 3.5 miles route was used during the data populating stage. The trail has a start/ending point at 3504-A Paramount Dr; stop point one at 3075 Sachem Ct S; stop point two at 1900 Foxglove Way. The researcher drove the testing vehicle from the start point, made a stop at stop point #1, then started going to stop point #2, and finally came back to the endpoint, resulting in a loop testing route. Figure 3.3 shows the testing route and all the waypoints on the map.

Table 3.6 All Test Cases Information

TC#	#1	#2	#3	#4	#5	#6
Test Vehicle	V#1	V#1	V#1	V#2	V#2	V#2
Test Device	D#1	D#1	D#2	D#1	D#2	D#2
Test App	A#2	A#3	A#2	A#2	A#1	A#2
Test Route	Y	N	Y	Y	Y	Y

3.5 Data Extraction

Six test cases were recorded on two mobile devices. Two sets of physical images were created with a designated forensics tool in total.

There was one forensic extraction approach when generating the mobile device image via the Cellebrite UFED Physical Analyzer. With the iOS device, the device was connected to the workstation via USB cable and followed the instruction by the UFED Physical Analyzer, the connected iOS device was jailbroken on the fly, and after the image was created successfully, the device was non-jailbroken. For the Android device, the OEM unlock function must be enabled before connecting to the workstation, so the Physical Analyzer can root the device during the acquisition.

When finished, a report was generated, which is ready for analysis using the Cellebrite UFED Reader.

Table 3.7 Detailed Information for Forensics Images

Image Number	Image #1	Image #2
Test Devices	iPhone X	Galaxy S6
Operating System	iOS	Android
Forensic Tool Used	Cellebrite	Cellebrite

CHAPTER 4. RESULTS

In this chapter, the summary of artifacts retrieved from the mobile devices after the use of target applications is presented. For the iOS device, Nonda ZUS and Mercedes-Benz Companion applications were analyzed. On the other hand, Nonda ZUS and FIXD were analyzed for the Android device. The retrieved artifacts include but are not limited to user account information, user vehicle information, OBDII adaptor information, vehicle diagnosis information, and geolocation, which contains trip and parking information.

4.1 iOS Device analyzing

In this subsection, the artifacts retrieved from the testing applications—Nonda ZUS and Mercedes-Benz Companion—installed on iOS devices are presented and discussed.

4.1.1 Nonda ZUS - iOS

This subsection focuses on the different types of artifacts that are recovered from the Nonda ZUS mobile application on the iOS device only. The file path for application package of Nonda ZUS is located at *Apple_iPhone*

X.zip/root/private/var/mobile/Containers/Data/Application/2406B96C-1BE5-4E8F-A379-8672923AB85E.

4.1.1.1 User Account Information

The username and sign-in email address can be found under the file path of *2406B96C-1BE5-4E8F-A379-8672923AB85E/Library/Preferences/com.zendesk.core.identity.plist*, which shows the username and user email are both “moblie.vehicle.forensics@gmail.com” as shown in

Figure 4.1. The password was not found under the application package file path mentioned in section 4.1.1.



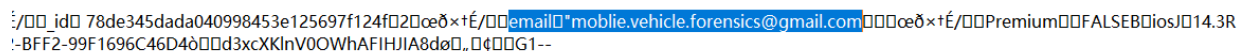
```

{"anonymous":{"name":"moblie.vehicle.forensics@gmail.com","email":"moblie.vehicle.forensics@gmail.com"}}

```

Figure 4.1 Username and email Address Part 1

From a different file path of *2406B96C-1BE5-4E8F-A379-8672923AB85E/Library/Application Support/Google/Measurement/google-app-measurement.sql*, a SQL file can be found. It records the email ID for Nonda ZUS and the number of vehicles that have been added to the account, as shown in Figure 4.2 and Figure 4.3.

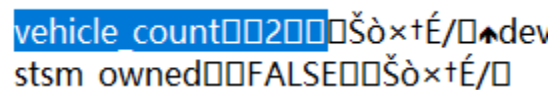


```

--BFF2-99F1696C46D4b3xcXKlnV0OWhAFIHJIA8d0,0400G1--

```

Figure 4.2 Username and email Address Part 2



```

vehicle_count2000Šò×tÉ/0♠dev
stsm_owned00FALSE00Šò×tÉ/0

```

Figure 4.3 User Vehicle Count

4.1.1.2 User Vehicle Information

From the previous subsection, the number of vehicles that have been linked to the account was recovered. A string called “vehicleId” can be found in the debug folder under the path of *2406B96C-1BE5-4E8F-A379-8672923AB85E/Documents/Debug*. Two unique vehicle IDs were found, reflecting the two testing vehicles used during the data population stage. The recovered vehicle IDs are: 082564a2881e48e59546d8552e8aec41, and

171c1fa08ffa4d9c8b70636207bde69e as shown in Figure 4.4 and Figure 4.5. The timestamp also matches the time recorded during the data population stage.

```
19:29:15 parking_engineStart_logic
19:29:15 trip_ignore_logic ["reason": "isTripRecording, ignore this start"]
19:29:15 vehicleStart ["vehicleId": "171c1fa08ffa4d9c8b70636207bde69e", "LocationPermissions":
```

Figure 4.4 User Vehicle ID Number Part 1

```
18:24:46 app_start
18:24:51 parking_engineStop_logic ["last5MinsVoltage": [], "vehicleId": "082564a2881e48e59546d8552e8aec41"]
18:24:51 vehicleStop ["vehicleId": "082564a2881e48e59546d8552e8aec41", "locationEnable": true, "parkingDevice": "obdlite",
```

Figure 4.5 User Vehicle ID Number Part 2

The vehicle ID recovered from the application should be generated by the Nonda ZUS rather than coming with the testing vehicles. However, the Vehicle Identification Number (VIN) was not recovered from the iOS device.

4.1.1.3 OBD II Adaptor Information

The model of the OBD II adaptor used during the data population stage was discovered from the same logfiles under the path of *2406B96C-1BE5-4E8F-A379-8672923AB85E/Documents/Debug*. The OBD II adaptor in the logfiles connected to the testing vehicle is called “obdlite,” as shown in Figure 4.6 and Figure 4.7. The model of the device was recorded every time when the mobile device and the adaptor established a connection via Bluetooth, no matter which vehicle was attached.


```

18:24:51 parking_engineStop_logic [{"last5MinsVoltage": [], "vehicleId": "082564a2881e48e59546d8552e8aec41"}]
18:24:51 vehicleStop [{"vehicleId": "082564a2881e48e59546d8552e8aec41", "locationEnable": true, "parkingDevice": "obdlite", "LocationP
18:24:56 data_upload [{"source": "uploadVoltageHistory"}]

```

Figure 4.6 OBD II Adaptor Model Information for Vehicle 1

```

19:35:10 parking_startLocationFunc_logic
19:35:10 vehicleStop [{"parkingDevice": "obdlite", "locationEnable": true, "LocationPermissions": "kCLAuthorizationStatusAuthorizedAlways", "vehicleId": "171c1fa08ffa4d9c8b70636207bde69e"}]

```

Figure 4.7 OBD II Adaptor Model Information for Vehicle 2

4.1.1.4 Vehicle Diagnostic Trouble Code Information

The safety scanning function was performed during the data population stage. The scanning process can also be recovered from the log files located at *Apple_iPhone X.zip/root/private/var/mobile/Containers/Data/Application/2406B96C-1BE5-4E8F-A379-8672923AB85E/Documents/Debug*, for both testing vehicles. According to the recovered log file, a safety scanning was performed on the testing vehicle with the ID of 171c1fa08ffa4d9c8b70636207bde69e at the time of 19:29:56 on 2021/10/07. The scanning was finished at 19:30:12, with zero DTC code detached, as shown in Figure 4.8.

```
19:30:12 [{"7E904410C0B18", "7EB04410C0B1C",  
010C  
010D  
0105  
0104  
19:30:12 safety_scan_result_without_dtc  
ATRV  
19:30:13 [{"7E904410C0ACC", "7EB04410C0ACC"  
010C  
010D  
0105  
0104  
ATRV  
19:30:13 [{"7E904410C0B00", "7EB04410C0AF8",
```

Figure 4.8 Vehicle DTC Scanning Log

4.1.1.5 Geolocation information

All the artifacts that include the geolocation data will be discussed under this subsection, including trip information, parking location, and geofence created by the iOS mobile application.

4.1.1.5.1 Trip information

The trip information can be recovered from the image, and there are different formats of artifacts to show it. First, under *Apple_iPhone*
X.zip/root/private/var/mobile/Containers/Data/Application/2406B96C-1BE5-4E8F-A379-8672923AB85E/Library/Caches/com.onevcate.Kingfisher.ImageCache.default, several .png files can be found, which shows the starting and the endpoint for each trip. The start point is identified as a green dot on the map, and the endpoint is identified as a red pin with a vehicle icon on the map which also acts as the parking location indicator, as shown in Figure 4.10 to Figure 4.15. By the use of Cellebrite Reader, the creation time of the png snapshot can be found. For example,

the “0a533abe923b945f870029a9292d6989” was created on 10/7/2021 at 7:43:05 PM(UTC-4) and last accessed on 10/14/2021 at 7:43:05 PM(UTC-4), which is the last time the application was opened as shown in Figure 4.9.

Date & Time	
Creation time	10/7/2021 7:43:05 PM(UTC-4)
Modify time	10/14/2021 7:43:05 PM(UTC-4)
Last access time	10/14/2021 7:43:05 PM(UTC-4)
Deleted time	
Change time	10/7/2021 7:43:04 PM(UTC-4)

Figure 4.9 File Information for Trip snapshot 0a533abe923b945f870029a9292d6989

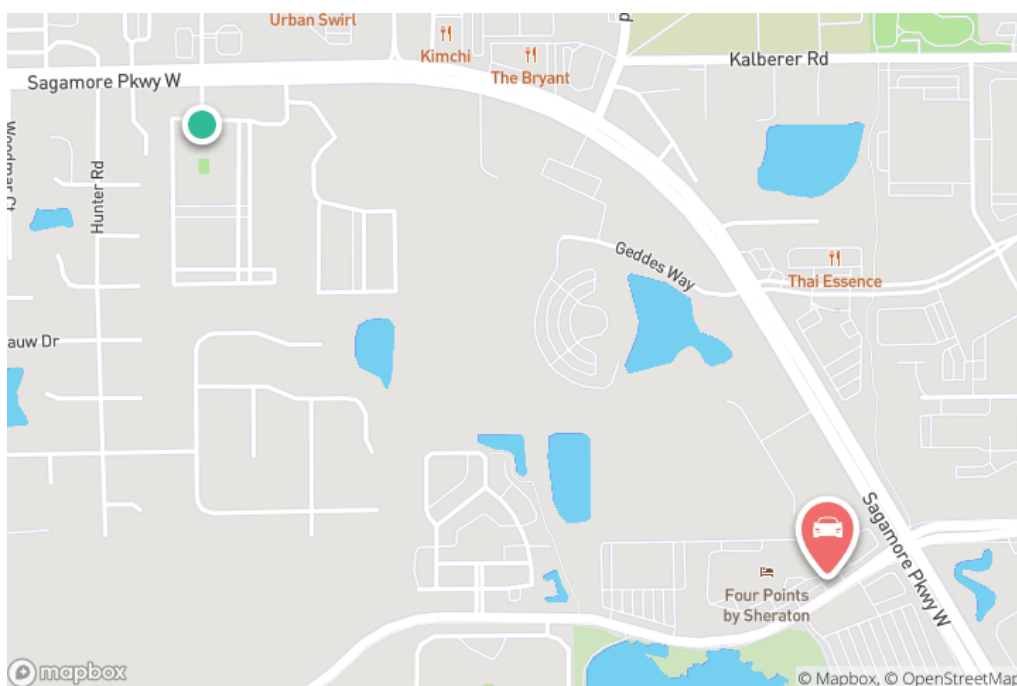


Figure 4.10 Trip snapshot 0a533abe923b945f870029a9292d6989

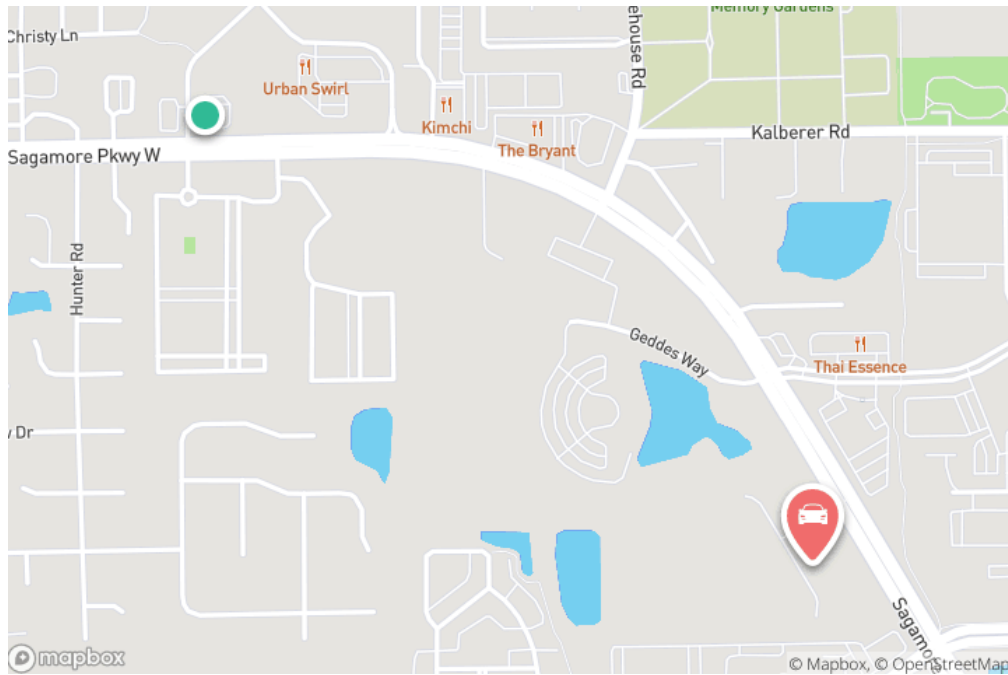


Figure 4.11 Trip snapshot 61b84eb8150fc1354f5a82057d32d350

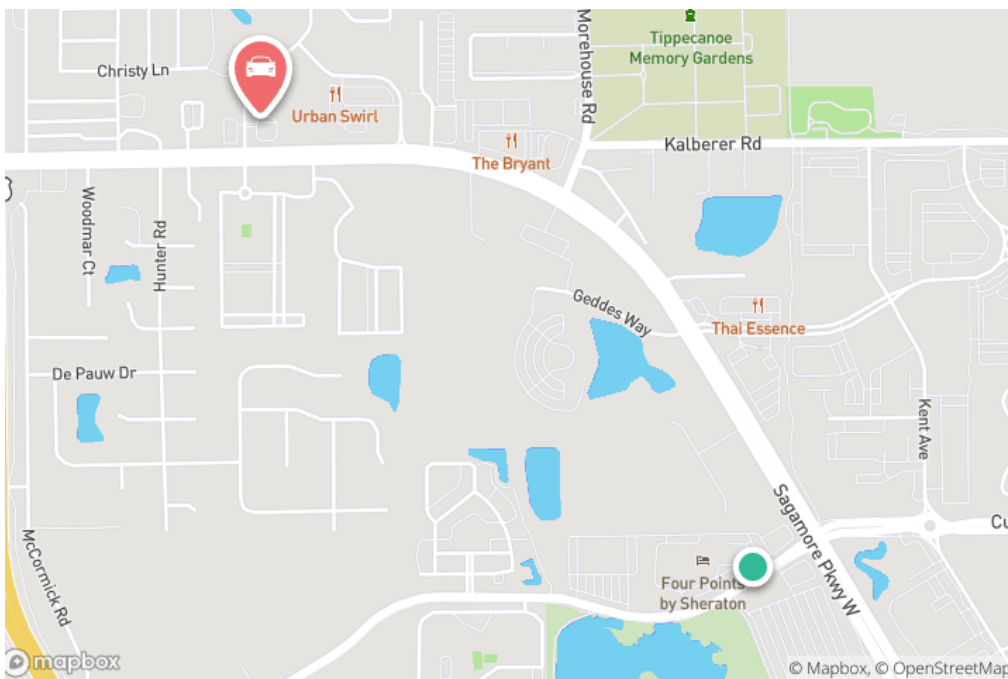


Figure 4.12 Trip snapshot 04914c36303dc1547b5c9ae2e08e3247

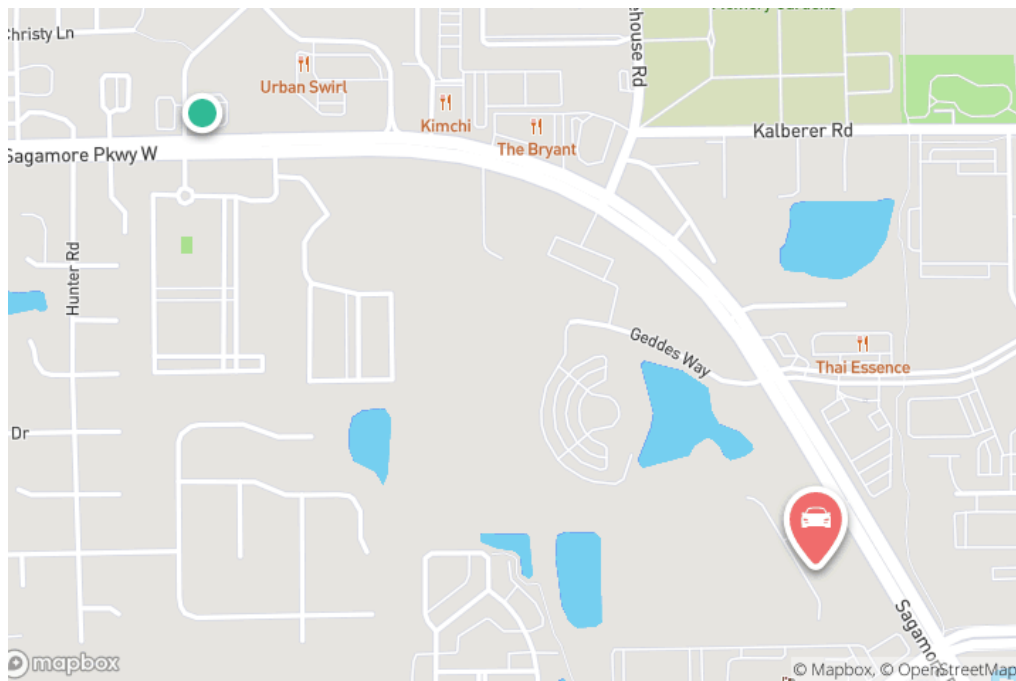


Figure 4.13 Trip snapshot b71577a3c670c14b403e60a37ab653b5

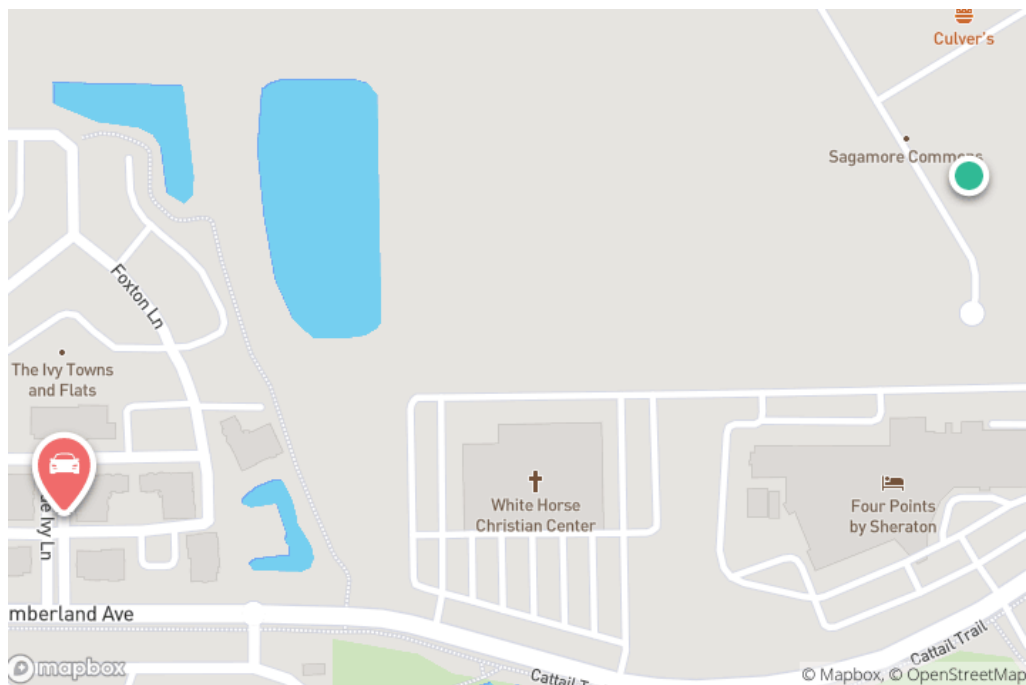


Figure 4.14 Trip snapshot bea57c38f837f495626d055711e6bfc2

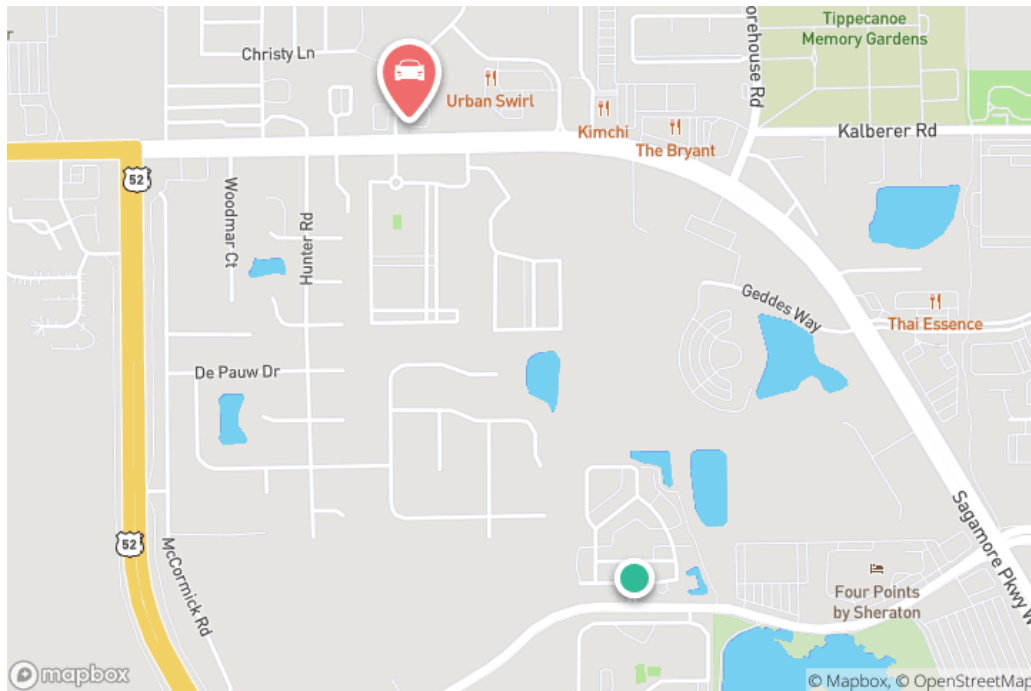


Figure 4.15 Trip snapshot db6143538dc868e47060685570a32cf1

4.1.1.5.2 Geofence information

The exact longitude and latitude were recorded in the system log files and motion activity log files. A geofence was created by the mobile application. The motion activities and GPS coordinates were also recorded in the log files, even if the mobile devices were not connected to the vehicle via Bluetooth. Two types of log files can be found under the path of *Apple_iPhone X.zip/root/private/var/mobile/Containers/Data/Application/2406B96C-1BE5-4E8F-A379-8672923AB85E/Documents/Debug*.

The motion activity logs started recording the current status by accessing the motion sensor on the mobile device. A sample of the activity log named *2021-10-04motionActivity.log* is shown in Figure 4.16. This log was generated in Chinese. The English translation annotations are also given in the same figure.

Current Status: Walking Accuracy: High, not recording
 17:40:20 当前状态: 步行 准确度: 高, 未记录 isDrive: 否 False
 17:40:26 当前状态: 步行 准确度: 高, 未记录, isDrive: 否
 Driving 17:42:58 当前状态: 驾车 准确度: 高, 未记录, isDrive: 是 True
 17:42:58 离开指定区域 40.47402587892904,-86.94490039729229, 140 -top
 17:42:58 开始后台打点 Left the designated area (Geofence)
 17:42:58 在圈外 Start to record the coordinates
 17:42:58 打点中 40.47402587892904,-86.94490039729229, 140 -top
 2021-10-04 21:42:58 +0000 Outside of the geofence
 Recording
 17:43:00 当前状态: 驾车 准确度: 高, 未记录, isDrive: 是

Figure 4.16 Motion Activity Log and Translation

When the mobile device connected to the OBDII adapter, the current status was updated to “driving” in the motion activity logs. The regular log file starts recording the connection status, including the vehicle ID that has been discussed in the previous subsections and the geolocation. The sample log of the *2021-10-04.log* indicates the format of the connection status and the geolocation, as shown in Figure 4.17 and Figure 4.18.

```

19:26:14 timeline_pv
19:26:16 settings_vehicle_settings
19:26:18 settings_vehicle_settings
19:26:20 settings_quick_add_obdlite_click
19:26:21 settings_add_obdlite_start ["vehicle_id": "171c1fa08ffa4d9c8b70636207bde69e"]
19:26:23 settings_add_obdlite_click_pair ["vehicle_id": "171c1fa08ffa4d9c8b70636207bde69e", "duration": 2]
19:26:51 settings_add_obdlite_fail ["reason": "device_not_found", "vehicle_id": "171c1fa08ffa4d9c8b70636207bde69e"]
19:26:55 settings_add_obdlite_click_pair ["duration": 34, "vehicle_id": "171c1fa08ffa4d9c8b70636207bde69e"]
19:27:23 settings_add_obdlite_fail ["vehicle_id": "171c1fa08ffa4d9c8b70636207bde69e", "reason": "device_not_found"]
19:27:25 settings_vehicle_settings
19:27:35 settings_add_obdpublic_start
19:28:24 ble_connect_succeed ["devicename": "OBDII"]
19:28:24 obdlite_connect
19:28:24 OBD-Public adapter state: OBD2AdapterState(rawValue: 3)
19:28:24 OBD-Public adapter state: OBD2AdapterState(rawValue: 4)

```

Figure 4.17 Geofence Information in the Application Log File

Name:
Description:
Type:
Origin:
Timestamp: 10/4/2021 10:43:39 PM
End time:
Position: (40.461592, -86.935424)
Aggregated locations:
Map Address:
Precision:
Confidence: 76
Map:
Category:
Source: ZUS - Save Car Expenses
Account:
Address:
Extraction: File System
Source file: Apple_iPhone X.zip/root/private/
var/mobile/Containers/Data/
Application/2406B96C-1BE5-4E8F
-A379-8672923AB85E/
Documents/
Debug/2021-10-04.log : 0xCCEF
(Size: 188115 bytes)

Map

Position: (40.461592, -86.935424)
Address:
Map Address:

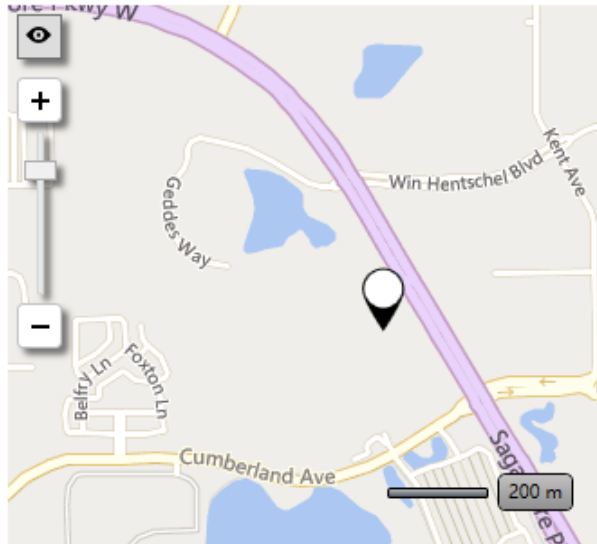


Figure 4.18 Geofence Information Captured by the Cellebrite UFED Reader

The parking location can also be found in the log files within the debug folder. The application recorded the parking location by recording the last geolocation before the mobile device disconnects to the OBD II adaptor, as shown in Figure 4.19. The number of parking counts made by the user was also stored in a SQL file under the file path of *Apple_iPhone X.zip/root/private/var/mobile/Containers/Data/Application/2406B96C-1BE5-4E8F-A379-8672923AB85E/Library/Application Support/Google/Measurement/google-app-measurement.sql* as shown in Figure 4.20.

Figure 4.19 Parking Information Found in the Application Log file

_e	211	211	1633650232.52909
_vs	370	370	1633650229.57884
timeline_pv	19	19	1633650229.57818
mileage_pv	36	36	1633650213.36222
obd_disconnect	8	8	1633650187.72086
obd_connect	9	9	1633650185.46567
obdlite_connect	11	11	1633650185.26075
ble_connect_succeed	11	11	1633650185.26046
app_enter_foreground	18	18	1633650183.41698
mileage_success	8	8	1633650183.19574
mileage_save	12	12	1633650157.87355
finder_parking_success	13	13	1633650153.3233
mileage_abandon	7	7	1633650153.31903
mileage_start	15	15	1633650136.23135
safety_scan_result_without_dtc	5	5	1633649847.09933
safety_scan_result_with_dtc_codes	10	10	1633649831.46512
safety_center_pv	8	8	1633649827.61387
obdlite_read_data_success_duration	8	8	1633649818.44731
obd_init_time	8	8	1633649818.30772
pid_support_response	9	9	1633649816.0465
phone_mileage_save	1	1	1633649730.87866
settings_pv	6	6	1633649440.89962
mileage_log_monthly_filter_confirm	1	1	1633649437.17057
sos_nonda_rsa_introduction_display	2	2	1633649420.85192
dashboard_customizable_pv	6	6	1633649363.96842

Figure 4.20 Parking Finder Count

4.1.2 Mercedes-Benz Companion iOS

The file path of Mercedes-Benz Companion mobile application is *Apple_iPhone X.zip/root/private/var/mobile/Containers/Data/Application/E393390A-4E0F-4A77-8CC1-5BA5F0107278*. Under the application package, as shown in Figure 4.21, only five artifacts were discovered. One database file was named *predictionmodule.db*. An analysis studio offline data file is named *data.data*. The other three files were Khronos texture (KTX) files. KTX is a type of image file introduced by Khronos Group(KTX Overview, 2022). Screenshots will be generated and stored as previews when users are switching between different applications on iOS devices (Khronos Texture, 2021).

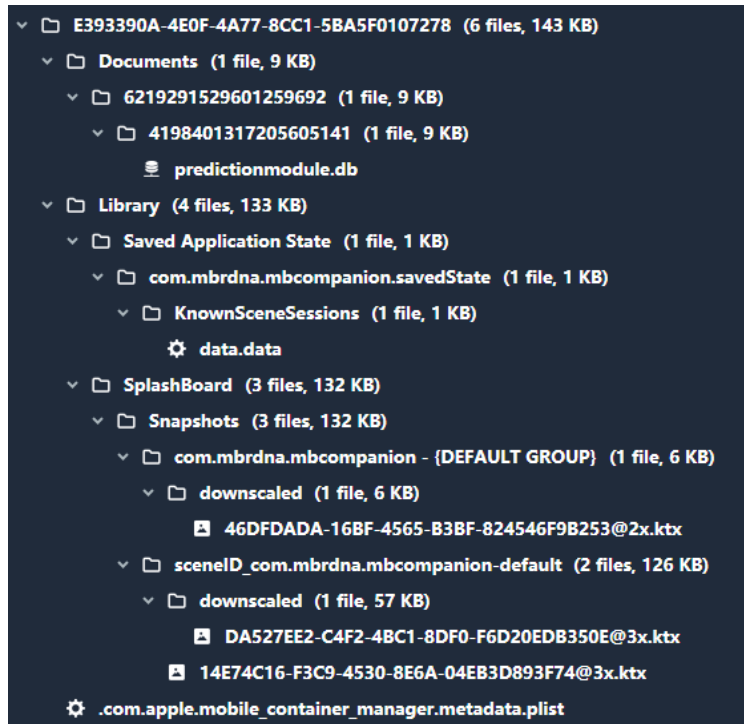


Figure 4.21 File System for MB-Companion Application Package

4.1.2.1 Database Files

Under the file path of *E393390A-4E0F-4A77-8CC1-5BA5F0107278/Documents/6219291529601259692/4198401317205605141/predictionmodule.db*, a database file was found. However, this file cannot be opened by the Cellebrite Reader. Alternative database browsers were used. The first browser was the SQLiteSPY 3.7.8, yet an error message box showed up, which stated the database file was encrypted or was not a database file, as shown in Figure 4.22.

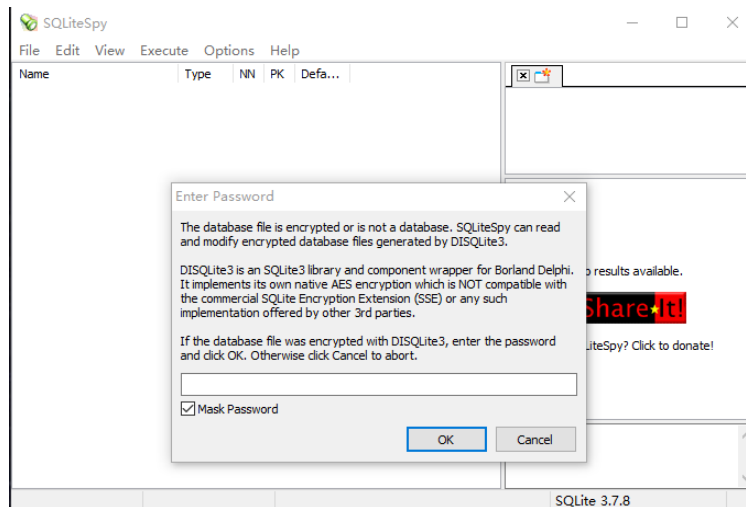


Figure 4.22 Error Message from SQLiteSPY

The second database was the DB Browser for SQLite Ver 3.12.2. However, a similar error message appeared on the screen, which indicated the message “Could not open database file. Reason: file is not a database,” as shown in Figure 4.23.

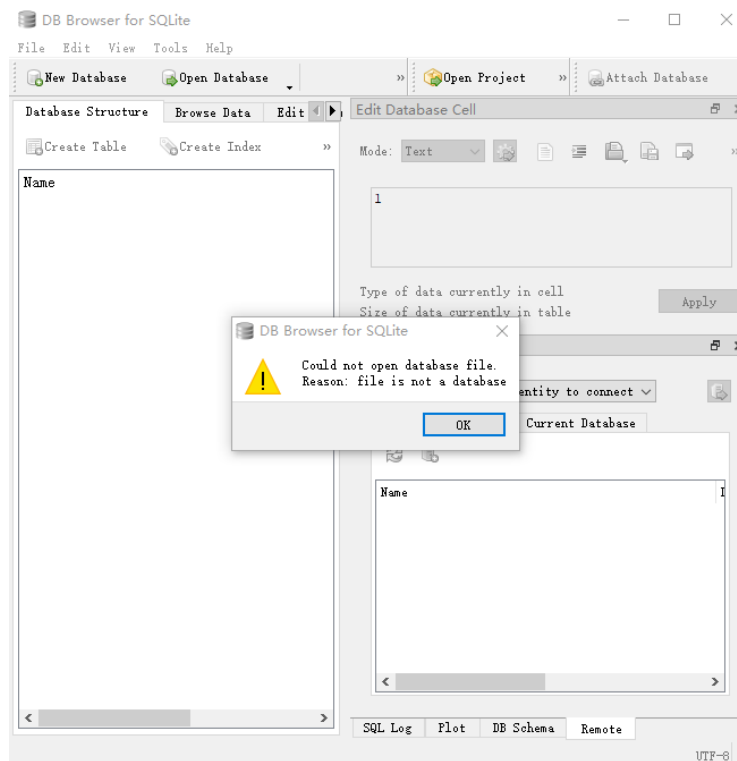
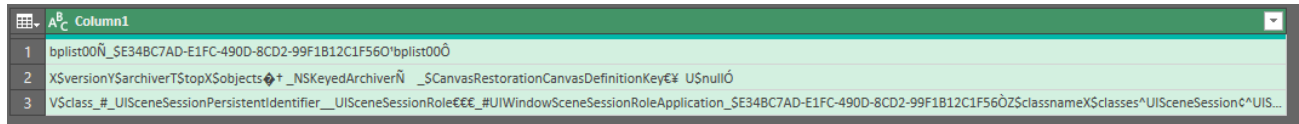


Figure 4.23 Error Message from DB Browser for SQLite

Another database file, *data.data*, could be accessed by using Microsoft Excel. Three records were recovered, as shown in Figure 4.24; such records do not contain artifacts that are related to the mobile device and vehicles.



	Column1
1	bplist00N_SE34BC7AD-E1FC-490D-8CD2-99F1B12C1F560*bplist00O
2	XSversionYSarchiverTStopXSobjects+ _NSKeyedArchiverN _SCanvasRestorationCanvasDefinitionKeyU\$ nullO
3	V\$class_#_UISceneSessionPersistentIdentifier__UISceneSessionRoleU\$ _#UIWindowSceneSessionRoleApplication_SE34BC7AD-E1FC-490D-8CD2-99F1B12C1F560Z\$classnameX\$classes^UISceneSessionUIS...

Figure 4.24 Records Found from data.data File

4.1.2.2 Khronos texture file

Three Khronos Texture (KTX) files are found in the application package, under the file path of *E393390A-4E0F-4A77-8CC1-*

5BA5F0107278/Library/SplashBoard/Snapshots/com.mbrdna.mbcompanion - {DEFAULT GROUP}/downscaled. The Cellebrite Reader was not able to process the KTX file. Therefore, a Mac computer was used to preview those KTX files, as shown in Figure 4.26. The first snapshot in Figure 4.26 only shows the logo of the application, and the second and the third snapshots reveal the parking location and the current location of the user. Moreover, the created time was recovered by checking the metadata of the KTX files, as shown in Figure 4.25.

DA527EE2-C4F2-4BC1-8D... x											
Image view	File Info										
Find:											
<div> <div>General</div> <table> <tr><td>Uid</td><td>501</td></tr> <tr><td>File size</td><td>58837 Bytes</td></tr> <tr><td>Chunks</td><td>1</td></tr> </table> </div>		Uid	501	File size	58837 Bytes	Chunks	1				
Uid	501										
File size	58837 Bytes										
Chunks	1										
<div> <div>Offsets</div> <table> <tr><td>Data offset</td><td>0x24F2393BC</td></tr> </table> </div>		Data offset	0x24F2393BC								
Data offset	0x24F2393BC										
<div> <div>Date & Time</div> <table> <tr><td>Creation time</td><td>10/7/2021 7:16:25 PM(UTC-4)</td></tr> <tr><td>Modify time</td><td>10/7/2021 7:16:25 PM(UTC-4)</td></tr> <tr><td>Last access time</td><td>10/7/2021 7:16:25 PM(UTC-4)</td></tr> <tr><td>Deleted time</td><td></td></tr> <tr><td>Change time</td><td>10/7/2021 7:16:25 PM(UTC-4)</td></tr> </table> </div>		Creation time	10/7/2021 7:16:25 PM(UTC-4)	Modify time	10/7/2021 7:16:25 PM(UTC-4)	Last access time	10/7/2021 7:16:25 PM(UTC-4)	Deleted time		Change time	10/7/2021 7:16:25 PM(UTC-4)
Creation time	10/7/2021 7:16:25 PM(UTC-4)										
Modify time	10/7/2021 7:16:25 PM(UTC-4)										
Last access time	10/7/2021 7:16:25 PM(UTC-4)										
Deleted time											
Change time	10/7/2021 7:16:25 PM(UTC-4)										

Figure 4.25 Metadata for KTX Image Files

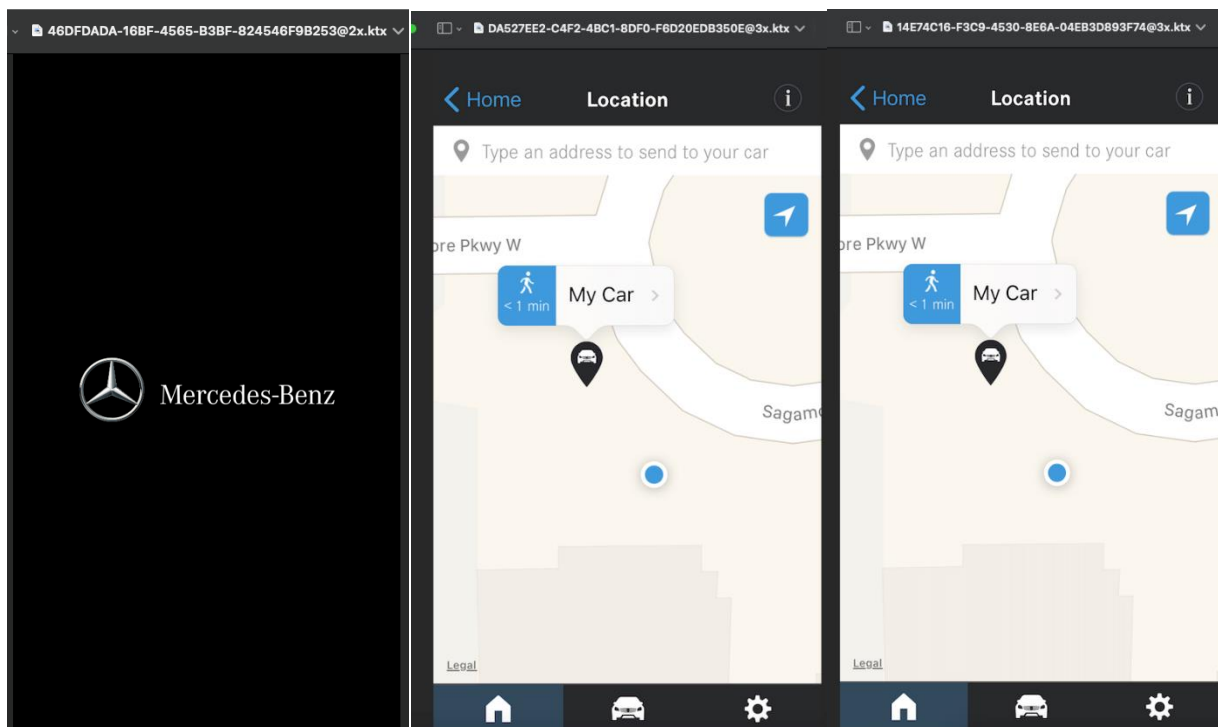


Figure 4.26 Converted KTX Image Files

4.2 Android Device Analyzing

This subsection will present and discuss the artifacts retrieved from the testing applications, Nonda ZUS and FIXD, installed on the Android device.

4.2.1 Nonda ZUS Android

This subsection concentrates on the different artifacts that are recovered from the Nonda ZUS mobile application on the Android device only. The types of artifacts include the user account information, user vehicle information, OBDII adaptor information, vehicle diagnosis information, and geolocation, including trip and parking information. The file path for the application package of Nonda ZUS: *Samsung CDMA_SM-G920P Galaxy*

S6.zip/sbin/.magisk/mirror/data/data/us.nonda.zus.

4.2.1.1 User Account Information

The user email address and generated user ID can be recovered from an XML file. The file path is *Samsung CDMA_SM-G920P Galaxy* *S6.zip/sbin/.magisk/mirror/data/data/us.nonda.zus/shared_prefs/UserSp.xml*. From this XML file, the login email is “moblie.vehicle.forensics@gmail.com”, and the backend user ID is “78de345dada040998453e125697f124f”. Both are the same as the recovered artifacts from the iOS device, as shown in Figure 4.27.

```

<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <int name="dashboard082564a2881e48e59546d8552e8aec41" value="0" />
  <int name="dashboard171c1fa08ffa4d9c8b70636207bde69e" value="0" />
  <string name="SP_KEY_LATEST_USER_EMAIL">moblie.vehicle.forensics@gmail.com</string>
  <string name="SP_KEY_LATEST_USER_ID">78de345dada040998453e125697f124f</string>
</map>

```

Figure 4.27 User Account Information from XML File

A realm database file is also recovered under the path of *Samsung CDMA_SM-G920P Galaxy S6.zip/sbin/.magisk/mirror/data/data/us.nonda.zus/files/realm-object-server/264f8463d37d87e3b974d3e7990b061c/fcks/zus.realm*. However, the realm file cannot be viewed by the Cellebrite reader. A MongoDB Realm Studio is needed to access the database. The user information, including the ID, username, email, phone number, username, gender, age, and other biographical information, is stored in this database file under the sub-table “UserDO,” as shown in Figure 4.28. However, the password is not discovered under the application package path.

id	username	email	phoneNumber	firstName	lastName	gender
string? (Primary Key)	string?	string?	string?	string?	string?	string?
78de345dada040998453e125697f124f	moblie.vehicle.forensics@gmail.com	moblie.vehicle.forensics@gmail.com		null	null	

Figure 4.28 User Account Information from realm Database

4.2.1.2 User Vehicle Information

The vehicle ID numbers that have been mentioned in the iOS result subsection can also be found in the Android application package. Under the file path of *Samsung CDMA_SM-G920P Galaxy S6.zip/sbin/.magisk/mirror/data/data/us.nonda.zus/shared_prefs*, two vehicle ID numbers are recovered. They are 082564a2881e48e59546d8552e8aec41 and

171c1fa08ffa4d9c8b70636207bde69e. These two ID numbers are the same ID numbers that have been recovered from the iOS device. However, one VIN starting with 55SWF8 that matches the ID of 171c1fa08ffa4d9c8b70636207bde69e is recovered in one XML file called *FILE_NAME_VEHICLE.xml* as shown in Figure 4.29. The Figure 4.30 and Figure 4.31 show only the backend vehicle ID in the XML files. This is the only VIN recovered from the Nonda ZUS mobile application for both iOS and Android devices.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="KEY_VEHICLE_VIN_CODE171c1fa08ffa4d9c8b70636207bde69e">55SWF8GE[REDACTED]</string>
  <string name="KEY_VEHICLE_ACTIVE_ID">171c1fa08ffa4d9c8b70636207bde69e</string>
  <int name="KEY_VEHICLE_PID_STATES171c1fa08ffa4d9c8b70636207bde69e" value="2" />
</map>
```

Figure 4.29 User Vehicle ID Information (with VIN)

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <float name="tpms_customize_pressure_low_front" value="193.0" />
  <float name="tpms_recommend_temp_front" value="70.0" />
  <float name="tpms_customize_pressure_high_rear" value="290.0" />
  <string name="vehicleId">082564a2881e48e59546d8552e8aec41</string>
  <float name="tpms_recommend_pressure_rear" value="241.0" />
  <float name="tpms_recommend_pressure_front" value="241.0" />
  <float name="tpms_customize_pressure_high_front" value="290.0" />
  <string name="tpms_pressure_alarm_mode">recommended</string>
  <float name="tpms_recommend_temp_rear" value="70.0" />
  <float name="tpms_customize_pressure_low_rear" value="193.0" />
  <string name="data_report_priority_device"></string>
</map>
```

Figure 4.30 User Vehicle ID Information (without VIN) Part 1

```

<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <float name="tpms_customize_pressure_low_front" value="193.0" />
  <float name="tpms_recommend_temp_front" value="70.0" />
  <float name="tpms_customize_pressure_high_rear" value="290.0" />
  <string name="vehicleId">171c1fa08ffa4d9c8b70636207bde69e</string>
  <float name="tpms_recommend_pressure_rear" value="241.0" />
  <float name="tpms_recommend_pressure_front" value="241.0" />
  <float name="tpms_customize_pressure_high_front" value="290.0" />
  <string name="tpms_pressure_alarm_mode">recommended</string>
  <float name="tpms_recommend_temp_rear" value="70.0" />
  <float name="tpms_customize_pressure_low_rear" value="193.0" />
  <string name="data_report_priority_device"></string>
</map>

```

Figure 4.31 User Vehicle ID Information (without VIN) Part 2

The users' vehicle information can also be found in the realm database file under the path of *Samsung data/data/us.nonda.zus/files/realm-object-server/264f8463d37d87e3b974d3e7990b061c/fcks/zus.realm*. The table stores all the vehicle information is called "VehicleDO," which includes the application back-end identification number, local identification number, owner's backend ID, odometers read, service mileage, make, model, VIN, built year, nickname, the timestamp for creation of the vehicle and last update, and the primary use of the vehicle as shown in Figure 4.32.

id string? (Primary Key)	localId string?	owner PublicUserDO?	odometers OdometerDO[]
082564a2881e48e59546d8552e8aec41	9685d56f-7030-4f62-b043-9b88ce107565	PublicUserDO [id = 78de345dada040998453e125697f124f]	[list of OdometerDO] 0
171c1fa08ffa4d9c8b70636207bde69e	171c1fa08ffa4d9c8b70636207bde69e	PublicUserDO [id = 78de345dada040998453e125697f124f]	[list of OdometerDO] 0

currentOdometer int	lastServiceMileage int	lastMaintenanceMileage int	lastMaintenanceDate string?	trim string?
59134779	0	0	1970-01-01	
11704	0	0	1970-01-01	

make string?	model string?	vin string?	year int	nickname string?
Audi	A3		2015	a3
Mercedes-Benz			2016	MB C Class

lastReplacingBattery int	lastReplacingBatteryMonth int	lastCh... int	lastCh... int	createdAt int	updatedAt int	valid bool
0	0	0	0	1633025036649	1633649731917	true
0	0	0	0	1633649077756	1634231675814	true

obdCompatibleUploaded bool	primaryUse string?	ownershipStatus string?	ownershipLength string?
true	To/From School	Owned	
true			

Figure 4.32 User Vehicle Detailed Information

The vehicle's configuration data is also in the same realm database. The table stores the configuration information called "VehicleConfigDO." The vehicle's backend identification number and some mobile application configurations were stored in this table, as shown in Figure 4.33.

vehicleId string? (Primary Key)	mileageCountryCode string?	mileageEnabled bool	mileageCustomDistanceUnit string?	mileageCustomRatePersonal double
082564a2881e48e59546d8552e8aec41	US	true	mile	0
171c1fa08ffa4d9c8b70636207bde69e	US	true	mile	0

Figure 4.33 User Vehicle Configurations Information

4.2.1.3 OBD II adaptor information

The history of the Bluetooth connection between the mobile device and the OBD II adaptor can be found in the log files under the file path of *Samsung CDMA_SM-G920P Galaxy S6.zip/sbin/.magisk/mirror/data/data/us.nonda.zus/files/ZUS/Log*. The adaptor information can also be found in the log file. According to the system logs, the process of searching ZUS-made Bluetooth devices was executed with the opening of the ZUS mobile application. The model of the OBD II adaptor can be recovered as obdlite (see Figure 4.35). Additionally, the device ID number is also shown in the log files as f180704e09ca40828ba2d62c94fa9c5f (see Figure 4.34). When the OBD II Bluetooth adaptor is attached to the vehicle and successfully connected to the mobile device, the vehicle's backend identification number is restored. From the *zus.realm* database file in the application package, two OBD II device records can be found. One device ID is f180704e09ca40828ba2d62c94fa9c5f. Another device ID is bc6454e9a75d4ea69673731d218d9449. However, there is only one OBD II adaptor used in the data population stage. A possible reason for the existence of the two device records is that it has been used on both testing vehicles. From the database record, every time the device connects to a new vehicle, a new device ID is created and memorized. Hence, according to the table called "DeviceDO" in the recovered realm database, device ID f180704e09ca40828ba2d62c94fa9c5f matches the device plugged into the vehicle that has the ID of 082564a2881e48e59546d8552e8aec41. Another device ID bc6454e9a75d4ea69673731d218d9449 matches the device attached with the vehicle that has ID of 171c1fa08ffa4d9c8b70636207bde69e. In the same table, the user ID, device model, created time, and unique identifier can also be found, as shown in Figure 4.36.

```

02 16:38:04      DeviceManagerImpl:syncAllDevice=1

02 16:38:04      restoreDevice:deviceDO=f180704e09ca40828ba2d62c94fa9c5f

02 16:38:04      performCreate:id=f180704e09ca40828ba2d62c94fa9c5f

02 16:38:04      Ble mac address is not available: 969C3AA1&FEC9&0C1E&3533&6820B94505F8

02 16:38:04      NBle init address=12:34:56:78:90:AB

```

Figure 4.34 OBD II Adaptor ID Number from Log Files

```

02 16:38:04      VehicleManagerImpl:restoreVehicle=171c1fa08ffa4d9c8b70636207bde69e

02 16:38:04      Vehicle:create=171c1fa08ffa4d9c8b70636207bde69e

02 16:38:04      onVehicleAdd:vehicle=171c1fa08ffa4d9c8b70636207bde69e fromRestore=true

02 16:38:04      N:TripEvent(type=obdlite, state=STOPPED)

02 16:38:04      VehicleStateManagerImpl:attach-merge-deviceCountChanges integer=0

```

Figure 4.35 OBD II Adaptor Model from Log Files

id string? (Primary Key)	localId string?	createdAt int	updatedAt int	userId string?	vehicleId string?	type string?
f180704e09ca40828ba2d62c94fa9c5f	722df423-b58f-40ad-8c4e-d3c6da2b23e7	1633387174538	1633387174538	78de345dada040998453e125697f124f	082564a2881e48e59546d8552e8a41	obdlite
bc6454e9a75d4ea69673731d218d9449	b13bcf67-8dcc-4d76-ba44-be3191d00365	1634230511409	1634230511409	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	obdlite

mHardwareDOList HardwareDO[]	identifier string?	boundAt int	valid bool	iccSendCoupon bool	firmware string?
[list of HardwareDO] 0	OBDI-uuid-000000222222-171c1fa08ffa4d	1634230511409	true	false	null
[list of HardwareDO] 0	OBDI-FFF0-969C3AA1&FEC9&0C1E&3533	1633387174538	true	false	null

Figure 4.36 OBD II Adaptor Detailed Information from the realm Database

4.2.1.4 Vehicle Diagnostic Trouble Code (DTC) Information

Unlike the ZUS application on an iOS device, the system logs only contain Bluetooth connection status. Hence the vehicle diagnostic data cannot be recovered in the log files. However, the DTC information can be found in the recovered *zus.realm* database file. Four vehicle safety scanning records are discovered under the table called “DtcCodeInfo.” In this table, the DTC code, title of the error code, and code-related information like description, symptoms, solutions, level of importance, and level of repairing difficulty are listed as shown in Figure 4.37.

code string?	title string?	desc string?
N0103	Maximum Historical Voltage Below Threshold	You can observe the health status of the battery through the battery health indicator.
N0101	Voltage Below Threshold	You can observe the health status of the battery through the battery health indicator.
N0103	Maximum Historical Voltage Below Threshold	You can observe the health status of the battery through the battery health indicator.
N0103	Maximum Historical Voltage Below Threshold	You can observe the health status of the battery through the battery health indicator.
causes string?	symptoms string?	solutions string?
· Continuous driving at low speeds causes the battery to not charge. The battery is aged or damaged.	· The battery's visual health indicator is black	Solution 1: Resume drivingThis solution car
· Continuous low-speed driving causes the battery to not charge. The battery is aged or damaged.	· The battery's visual health indicator is black	Solution 1: Resume drivingThis solution car
· Continuous driving at low speeds causes the battery to not charge. The battery is aged or damaged.	· The battery's visual health indicator is black	Solution 1: Resume drivingThis solution car
· Continuous driving at low speeds causes the battery to not charge. The battery is aged or damaged.	· The battery's visual health indicator is black	Solution 1: Resume drivingThis solution car
cost string?	repair_importance_level string?	repair_difficulty_level string?
	MediumSome issues will more than likely fi	ModerateRepair could require replacing a component
If the battery needs to be replaced. Replaci	MediumSome issues will more than likely fi	ModerateRepair could require replacing a component
	MediumSome issues will more than likely fi	ModerateRepair could require replacing a component
	MediumSome issues will more than likely fi	ModerateRepair could require replacing a component

Figure 4.37 Four Records of the Result from Vehicle Safety Scanning

Moreover, another table in the *zus.realm* database called “DtcDO” also has the result of the vehicle safety scan. Two DTC records were recovered. This table includes data fields like DTC ID, DTC error code, timestamp of getting the error code, the status of appointing, the mileage when this code was occurring, the ID of the vehicle, and code status (see Figure 4.38).

id string? (Primary Key)	code string?	occurredAt int	isAppointed bool
a4924aa11d62486ba1ae27993c8dcf53	N0103	1635897012459	false
5810b12373bc4e7c8ecfb0d66e414667	N0101	1635893440589	false

isAppointed bool	deviceId string?	miles int	scheduleTime string?
false	bc6454e9a75d4ea69673731d218d9449	0	
false	bc6454e9a75d4ea69673731d218d9449	0	

status string?	vehicleId string?	is_help... int	dtcCodeInfo DtcCodeInfo?
not_reserved	171c1fa08ffa4d9c8b70636207bde69e	0	DtcCodeInfo
not_reserved	171c1fa08ffa4d9c8b70636207bde69e	0	DtcCodeInfo

Figure 4.38 Two DTC Records with Detailed Info

4.2.1.5 Geolocation information

All the artifacts that include the geolocation data will be discussed in this subsection, including trip information, parking location, and geofence created by the Android mobile application.

4.2.1.5.1 Trip information

First of all, similar to the iOS mobile application, a total of five snapshots in .png format of the trips can be recovered under the file path of *Samsung CDMA_SM-G920P Galaxy S6.zip/sbin/.magisk/mirror/data/data/us.nonda.zus/cache/image_manager_disk_cache*. The starting point was marked as the green point on the map, and the end point of the trip was the red pin with a vehicle icon on the map (see Figure 4.40). The snapshots can only show the start and end points of the trip, yet the specific route and GPS coordinates are not available. Also, by using Cellebrite Reader, the modify time of each snapshot shows the trip end time by examining the modify time. For example, the snapshot

4c1704a21b94871c1d1661b0137b4e6dd3bd296f1cb96d514129d7d39103930b.0 was modified at 11/2/2021 7:54:37 PM(UTC-4) as shown in Figure 4.39 and Figure 4.40.

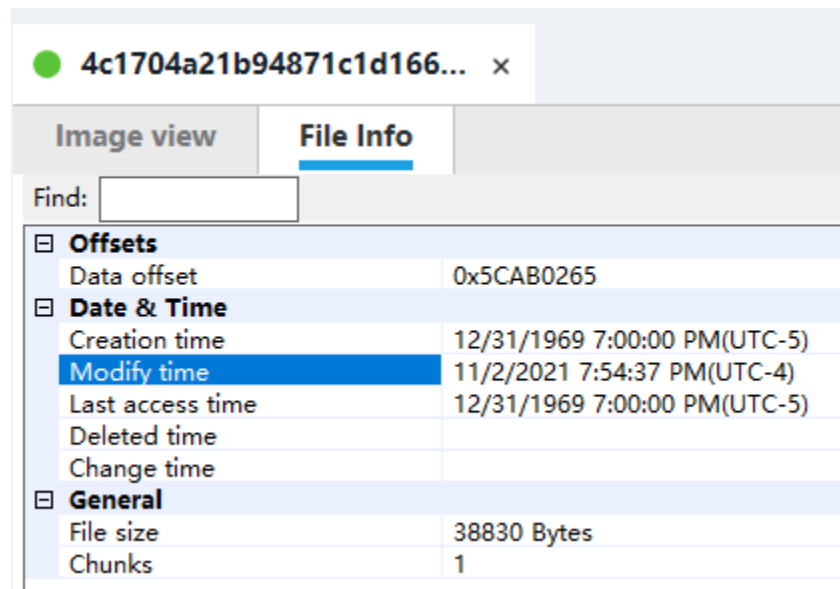


Figure 4.39 File Information for Trip Snapshot
4c1704a21b94871c1d1661b0137b4e6dd3bd296f1cb96d514129d7d39103930b.0

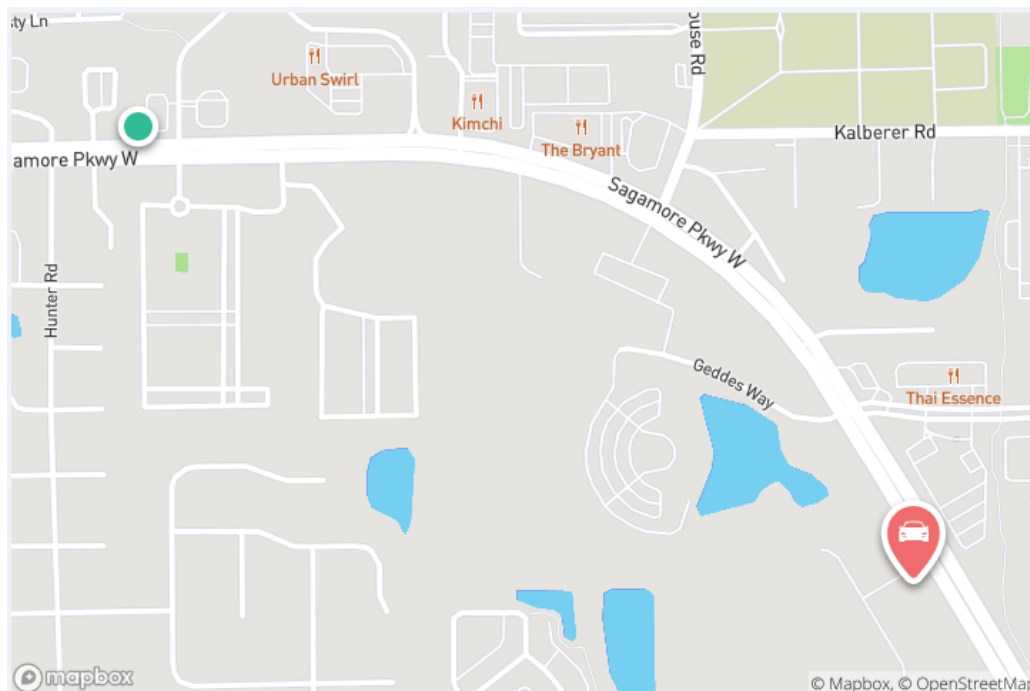


Figure 4.40 Trip Snapshot
4c1704a21b94871c1d1661b0137b4e6dd3bd296f1cb96d514129d7d39103930b.0

Secondly, a much-detailed trip dataset can be found under the *zus.realm* database. One of the tables is called “TripDO,” which contains application-generated data and user-entered information. The data generated from the application includes local trip ID, trip ID, device ID, the ID of the vehicle used for the trip, user ID, latitude and longitude of the start point, the street address of the start point (estimated by the application), the timestamp when the trip started, latitude and longitude of the end point, the street address of the end point (estimated by the application), the timestamp when the trip ended, total distance was driven, the timestamp for trip creation and update, cashback value, and the URL of the trip image (all trip images can be accessed by copy and paste to any internet browser). The user-entered information encompasses parking fee value, toll fee value, the type, the purpose, and the note for the trip. A total of seventeen records of trip information can be found in this table, as shown in Figure 4.41, Figure 4.42, and Figure 4.43.

localId string? (Primary Key)	id string?	deviceId string?	vehicleId string?
171c1fa08ffa4d9c8b70636207bde69e1635894749124	6bab1ffa332d4813bb659c81c5f3078f	null	171c1fa08ffa4d9c8b70636207bde69e
171c1fa08ffa4d9c8b70636207bde69e1634230626222	c0e3855429594dad4e0f3f4b991f16b	null	171c1fa08ffa4d9c8b70636207bde69e
171c1fa08ffa4d9c8b70636207bde69e1635897058891	6d95f8e988c64785b3a17225fea420d4	null	171c1fa08ffa4d9c8b70636207bde69e
082564a2881e48e59546d8552e8aec411633649457382	17f7c45be5c441f58b685f22cfc9ecb4	null	082564a2881e48e59546d8552e8aec41
171c1fa08ffa4d9c8b70636207bde69e1633649078000	d49a8c52874c4e7e9ee95400bd5fc456	null	171c1fa08ffa4d9c8b70636207bde69e
082564a2881e48e59546d8552e8aec411633384993205	206205f5f4164d13bcd5d13bdd55a4aa7	null	082564a2881e48e59546d8552e8aec41
082564a2881e48e59546d8552e8aec411633387178000	27dcd55074624266bc7c8ad6e31a52ca	null	082564a2881e48e59546d8552e8aec41
171c1fa08ffa4d9c8b70636207bde69e1635894004424	3380c09f8bf640f8b1d002c75fda60b0	null	171c1fa08ffa4d9c8b70636207bde69e
171c1fa08ffa4d9c8b70636207bde69e1634231390677	1f77e1699d0747c0b3e1e8b05445c30f	null	171c1fa08ffa4d9c8b70636207bde69e

userId string?	startLat double	startLng double	startLocationName string?	startedAt int
78de345dada040998453e125697f124f	40.4668506	-86.9546951	3457, Bethel Drive, West Lafayette	1635894749124
78de345dada040998453e125697f124f	40.4685858	-86.9479768	3503 Paramount Drive, West Lafayette, IN	1634230626222
78de345dada040998453e125697f124f	40.4678839	-86.9479184	2220, Sagamore Parkway West, West Lafay	1635897058891
78de345dada040998453e125697f124f	40.466793431006806	-86.94733249031654	2243 Sagamore Pkwy W	1633649457382
78de345dada040998453e125697f124f	40.46680956617041	-86.9472485874658	Sagamore Pkwy W	1633649078000
78de345dada040998453e125697f124f	40.4680556	-86.9469159	3503, Paramount Drive, West Lafayette	163384993205
78de345dada040998453e125697f124f	40.46806148722811	-86.94691431716731	3503 Paramount Dr	1633387178000
78de345dada040998453e125697f124f	40.4729635	-86.9447442	3879 Ledyard Street, West Lafayette, IN 47	1635894004424
78de345dada040998453e125697f124f	40.4598694	-86.9420646	1933, Foxglove Way, West Lafayette	1634231390677

Figure 4.41 Detailed trip Information from realm database part 1

endLat double	endLng double	endLocationName string?	endedAt int	parkingFee double
40.4730358	-86.944809	3879, Ledyard Street, West Lafayette	1635895064500	0
40.4619884	-86.9355683	3001 Sagamore Parkway West, West Lafayett	1634230840618	5
40.4621869	-86.9352124	S. Schem Boulevard, West Lafayette	1635897211000	0
40.459887916259895	-86.93469660364737	1600 Cumberland Ave	1633649731049	0
40.45990254268093	-86.93468218677391	1600 Cumberland Ave	1633649710000	15
40.461622	-86.9354399	S. Schem Boulevard, West Lafayette	1633385310000	15
40.46159212481273	-86.93542465575685	Sagamore Pkwy W	1633387426000	15
40.4669221	-86.9547503	3457 Bethel Drive, West Lafayette, IN 47906	1635894732087	0
40.4666826	-86.9469944	2243, Sagamore Parkway West, West Lafayet	1634231674475	10
40.467997500405126	-86.94692908465063	3503 Paramount Drive, West Lafayette, IN 47	1633388203000	5
40.4680507	-86.9468675	3503, Paramount Drive, West Lafayette	1633386401000	0
40.459795	-86.9421273	1933, Foxglove Way, West Lafayette	1635897680000	0
40.4603807	-86.9426349	212, Foxglove Way, West Lafayette	1635893892866	0
40.459818978806716	-86.94161489337662	Blue Ivy Ln	1633387783000	5
40.4600269	-86.9421859	213, Foxglove Way, West Lafayette	1634231228408	12
40.46833741948052	-86.94685279399803	3503 Paramount Drive, West Lafayette, IN 47	1633650153000	5
40.4730396	-86.9448489	3879, Ledyard Street, West Lafayette	1635898159000	0

distance double	note string?	tolls double	type string?	purpose string?
1459.949951171875	<i>null</i>	0	Business	
1751.7900390625	having lunch	2	Business	Meal / Entertain
1729.1199951171875	<i>null</i>	0	Business	Meal / Entertain
1861.3478859005418		0		
1882.0336551934215	Go to dinner	25	Business	Meal / Entertain
1565.9599609375	this is a trip to restaurant	20	Business	Meal / Entertain
1617.5868109600908	Getting dinner	20	Business	Meal / Entertain
4046.97998046875	<i>null</i>	0	Business	
2897.52001953125	back to home	5	Personal	Commute
3088.8529298770104	Back to home from the visiting	0	Personal	Commute
2912.070068359375	back to home	20	Personal	Commute
2179.1201171875	<i>null</i>	0	Business	Meeting
1964.7099609375	<i>null</i>	0	Business	Meeting
1266.389895631897	Visit a friend	0	Personal	Commute
1465.0999755859375	visiting friends	5	Personal	Commute
3709.2338710400586	Go to. The bank	10	Personal	Commute
3544.719970703125		0		

Figure 4.42 Detailed trip Information from realm database part 2

valid bool	createdAt int	updatedAt int	value float	potentials PotentialDO[]	autoGen bool	uploaded bool
false	1635895068130	1635897276405	0.5080146789550781	[list of PotentialDO]	4	true
true	1634230970575	1634231312966	0.6095654964447021	[list of PotentialDO]	4	true
false	1635897254722	1635897313627	0.6016770601272583	[list of PotentialDO]	4	true
true	1633649731904	1633649731904	0	[list of PotentialDO]	4	true
true	1633649731511	1633649794395	0.65488600730896	[list of PotentialDO]	4	true
true	1633385397268	1633385824097	0.5449026823043823	[list of PotentialDO]	4	true
true	1633387426445	1633387466941	0.5628671646118164	[list of PotentialDO]	4	true
false	1635895063574	1635897274874	1.4082162380218506	[list of PotentialDO]	4	true
true	1634231675803	1634231714979	0	[list of PotentialDO]	4	true
true	1633388207762	1633388260609	0	[list of PotentialDO]	4	true
true	1633386466344	1633386508137	0	[list of PotentialDO]	4	true
false	1635897742619	1635897790371	0.7582623362541199	[list of PotentialDO]	4	true
false	1635893894457	1635893943248	0.6836546063423157	[list of PotentialDO]	4	true
true	163338784353	1633387846738	0	[list of PotentialDO]	4	true
true	1634231229112	1634231279029	0	[list of PotentialDO]	4	true
true	1633650183004	1633650212654	0	[list of PotentialDO]	4	true
true	1635898207489	1635898207489	0	[list of PotentialDO]	0	true
trip_img_url string?						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9546951,40.4668506),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9479768,40.4685858),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9479184,40.4678839),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.94733249031654,40.466793431006806),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_locati						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9472485874658,40.46680956617041),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9469159,40.4680556),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.94691431716731,40.46806148722811),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_locatio						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9447442,40.4729635),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9420646,40.4598694),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.94161494270614,40.45980405531866),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_locatio						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9411559,40.4598511),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9354098,40.4619147),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9353934,40.4620734),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.93538484171678,40.461585419290195),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_locati						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.9352348,40.4621416),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_location_icon2x.png(-86						
https://api.mapbox.com/styles/v1/mapbox/streets-v11/static/url-https%3A%2F%2Fimg.nonda.co%2F1%2Fgreen_point2x.png(-86.93467732527007,40.45985237699045),url-https%3A%2F%2Fimg.nonda.co%2F1%2Fcar_locatio						

Figure 4.43 Detailed trip Information from realm database part 3

However, given the undetailed trip information, the driver’s exact route driven still remains unknown if only using this “TripDO” table. Another table called “LocalGpsDO,” also from the *zus.realm* database, is believed to fill this vacancy. In this table, a new record was created every specific time, containing well-detailed GPS information. Such data includes record ID, user ID,

vehicle ID, device ID, timestamp, longitude, latitude, altitude, instantaneous speed, instantaneous acceleration, and uploading status, as shown in Figure 4.44. Cross-examination can be performed based on the “TripDO” table and this “LocalGpsDO” table from the timestamp. The geolocation from the “LocalGpsDO” table can be used to fill out the vacancy between the start and end points from the “TripDO” table. A total of 2361 GPS locations can be recovered from this table.

id string? (Primary Key)	userid string?	vehicleid string?	deviceid string?
171c1fa08ffa4d9c8b70636207bde69e1635893559056	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893559489	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893559963	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893560457	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893561447	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893562471	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893563452	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449
171c1fa08ffa4d9c8b70636207bde69e1635893564456	78de345dada040998453e125697f124f	171c1fa08ffa4d9c8b70636207bde69e	bc6454e9a75d4ea69673731d218d9449

time int	lng double	lat double	alt double	spd float	brg float	acc float	uplc bool
1635893559056	-86.9353934	40.4620734	176.1008952121409	0.8308705687522888	109.98469543457031	7.3130002021789!	true
1635893559489	-86.9353892	40.4620706	175.97580110484088	0.7631720304489136	113.83924865722656	5.7960000038146!	true
1635893559963	-86.9353868	40.4620695	176.11917525652777	0.7581964135169983	113.98493194580078	6.1640000343322'	true
1635893560457	-86.9353848	40.4620682	176.08633788971784	1.1120457649230957	105.93134307861328	5.9629998207092!	true
1635893561447	-86.9353748	40.4620657	176.1252521478296	0.5163935422897339	102.95458221435547	6.1640000343322'	true
1635893562471	-86.9353787	40.4620664	176.08132035427084	0.44304776191711426	104.80347442626953	5.4860000610351!	true
1635893563452	-86.9353843	40.4620637	176.09872171428069	0.412250816822052	105.07418060302734	5.3520002365112!	true
1635893564456	-86.9353975	40.4620564	175.86166301527817	1.7411867380142212	233.4776153564453	5.4860000610351!	true

Figure 4.44 Detailed GPS Information from the realm database

For example, the trip with ID of 3f38e4694b6d411988497a183bd10ae1 can be found in the “TripDO” table. The start point was at the location of S, Sachem Boulevard, West Lafayette, GPS coordinates of (40.4620734, -86.9353934). The end point was at the location of 212 Foxglove Way, West Lafayette, with GPS coordinates of (40.4603807, -86.9426349). The trip was started at the timestamp 1635893558802 to 1635893892866. The snapshot generated by the mobile application is shown in Figure 4.45.

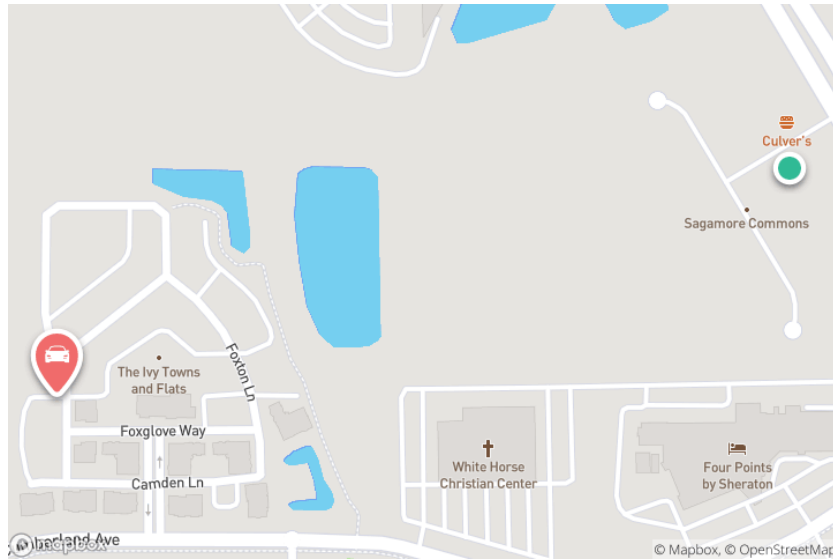


Figure 4.45 Trip snapshot for trip ID 3f38e4694b6d411988497a183bd10ae1

Then, the GPS coordinates recorded between the timestamps can be found in the “LocalGpsDO” table by sorting the records by time. A total of 298 records can be recovered. Thirty-two coordinates are picked among them to draw the waypoints using an online GPS waypoint planner tool called Geoplanner V3.1. It indicates the exact route can be recovered as well. The recreated route map is shown in Figure 4.46.

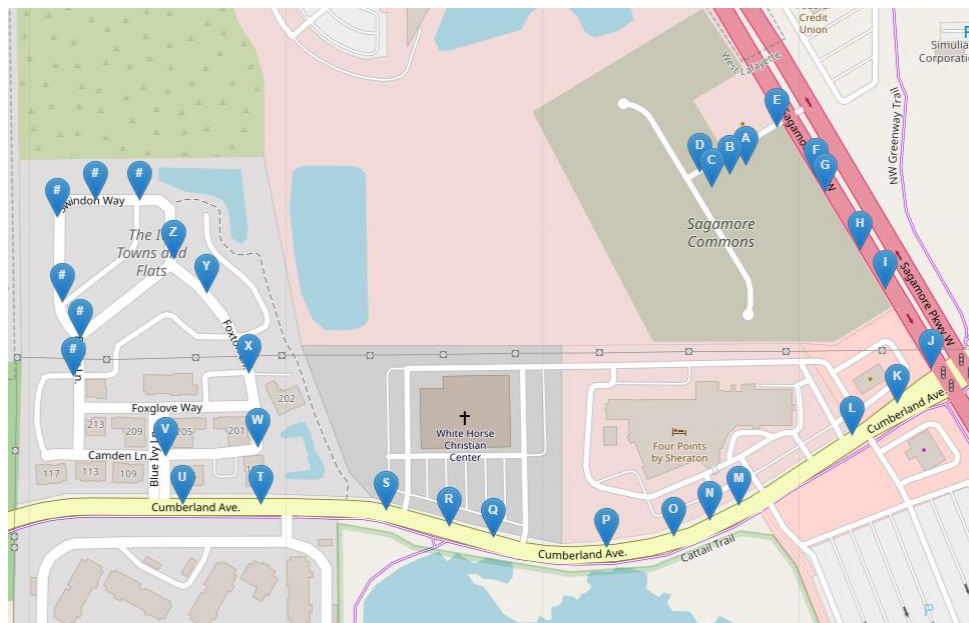


Figure 4.46 Recreated route map for trip ID 3f38e4694b6d411988497a183bd10ae1

4.2.1.5.2 Geofence Information

Unlike the iOS ZUS mobile application, no motive activity log can be found. However, a table named “GeofenceDO” is found in the *zus.realm* database. In this table, four records of geofence were recovered. The data stored in this table embraces the geofence ID, latitude and longitude, radius, event, expiration duration, create time, and vehicle ID, as shown in Figure 4.47.

geofenceld string (Primary Key)	latitude double	longitude double	radius float
-86.9481-300MI	40.4687218	-86.9480782	300
-86.9481	40.4687218	-86.9480782	100
-86.9448-300MI	40.4729763	-86.9447561	300
-86.9448	40.4729763	-86.9447561	100

event int	expirat... int	createTime int	createType int	vehicleId string
2	-1	1635896972875	0	171c1fa08ffa4d9c8b70636207bde69e
2	-1	1635896972875	0	171c1fa08ffa4d9c8b70636207bde69e
2	-1	1635898204293	2	171c1fa08ffa4d9c8b70636207bde69e
2	-1	1635898204293	2	171c1fa08ffa4d9c8b70636207bde69e

Figure 4.47 Geofence Information from the realm database

4.2.1.5.3 Parking information

The parking information can be recovered from the *zus.realm* database; a table named “LastParokingDO” is where the parking information is stored. Data such as local ID, user ID, vehicle ID, the timestamp for parking date/time, GPS coordinates, accuracy, device info, image, system estimated parking street address, and the status of synchronizing are contained as shown in Figure 4.48.

localId string? (Primary Key)	id string?	user PublicUserDO?
1eec3aea011e48f2943d96a013df8f73	1eec3aea011e48f2943d96a013df8f73	PublicUserDO {id = 78de345dada040998453e125697f124f}
2afa87ba512a4faca40b536eb8fb647	2afa87ba512a4faca40b536eb8fb647	PublicUserDO {id = 78de345dada040998453e125697f124f}

vehicleId string?	parkingAt int	lat double	lng double
082564a2881e48e59546d8552e8aec41	1633650133226	40.468156579919594	-86.94685924806348
171c1fa08ffa4d9c8b70636207bde69e	1635898204300	40.4730396	-86.9448489

accuracy float	device DeviceDO?	image UploadFileDO?	address string?	isSynced bool
5	null	UploadFileDO {url = }	3503, Paramount Drive, West Lafayette	true
5.111000061035156	null	null	3879, Ledyard Street, West Lafayette	true

Figure 4.48 Parking Information from the realm database

4.2.2 FIXD Android

The file path of the FIXD mobile application is *Samsung CDMA_SM-G920P Galaxy S6.zip/sbin/.magisk/mirror/data/data/com.fixdapp.two*. Unlike the Nonda ZUS, FIXD can only perform a DTC safety scanning. The application provides no other functions. Therefore, the investigation only focuses on the artifacts related to the vehicle and the scanning result.

4.2.2.1 User Account Information

An XML file can be discovered under the file path of *Samsung CDMA_SM-G920P Galaxy S6.zip/sbin/.magisk/mirror/data/data/com.fixdapp.two/shared_prefs/FIXD.xml*. The users' email accounts can be discovered in this XML file (see Figure 4.49).

```
<string name="MixpanelAlreadyAllasedUsersKey">["411a4615-3c60-4489-85be-15f7adb74508"]</string>
<string
name="user_repository_key">{"id":"2067813","name":"","email":"","mobile_vehicle_forensics@gmail.com","authentication_token":"","7gig3hMXh
hJ5wVr.6f0ojrD3f/IwprR8AkBQqRsdzhkz6sKFF1ygra6fDvGG/Ex2v3z9P0LYfol1uczB","universal_uuid":"","411a4615-3c60-4489-85be-15f7adb74508","sensor_id":null,"phone_
number":null,"diy_level":null,"primary_organization_id":null,"default_location_id":null}</string>
<string name="SENSOR_STORAGE_KEY">{"id":"2458549","name":"","CLASS":"","code":"","CY009635","user_id":"2067813","device_unique_id":"","38a3eca4-6e4e-47d7-99f0-6891fff450ed","mac_address":"","66:1B:
11:72:20:33","discovered_at":"","2021-11-02T22:42:52.329Z"}</string>
```

Figure 4.49 User email address from the XML file

4.2.2.2 Vehicle information

In the same *FIXD.xml* file, the last connected vehicle VIN that starts with WAUBFGFF can be found, which is the second testing vehicle. The time of the last connection activity can also be found in the file, as shown in Figure 4.50.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="LAST_PULL_TIME_INSTANT_KEY">&quot;2021-11-02T23:46:07.183Z&quot;</string>
  <string name="migration_version">2021-07-16</string>
  <string name="SELECTED_VEHICLE_KEYS">{&quot;vehicleId&quot;;2882962,&quot;vin&quot;;&quot;WAUBFGFF[REDACTED]&quot;;}</string>
  <string name="GOOGLE_PLAY_PURCHASED_SKUS_KEY">[]</string>
  <string name="Sell Page Clarity">&quot;Clearest&quot;</string>
  <string
```

Figure 4.50 Vehicle's VIN Information from the XML file

A search was explored by using the first testing vehicle's VIN. However, no data were found in the application package. Moreover, No vehicle DTC record was able to be recovered from the Android advanced logic image.

CHAPTER 5. DISCUSSION AND FUTURE WORK

5.1 Research Question

This study aims to answer the question of what forensically relevant artifacts can be recovered from the MB Companion, FIXD, and Nonda ZUS mobile applications on iOS and Android devices. A comprehensive overview is given in Chapter 4, which includes the artifacts successfully recovered from the investigated mobile applications on both iOS and Android devices. Designated sub-research questions are answered below.

5.1.1 Research Question 1

The first research question was, “Will complete route information be pictured by the geolocation data recovered from MB-Companion and Nonda ZUS?” As depicted in Table 5.1, it was possible to recreate the route using the geolocation data recovered but only for the Nonda ZUS application.

Table 5.1 Comparison of geolocation related artifacts found from both iOS and Android devices

	MB-Companion	ZUS-iOS	ZUS-Android
Geolocation Info	NO	YES	YES
Trip Info	NO	YES	YES
Detailed GPS Data	NO	PARTIAL	YES

5.1.2 Research Question 2

The second research question was, “What different vehicle information can be recovered from the MB Companion, FIXD, and Nonda ZUS applications?” As presented in Table 5.2, more

vehicle-related artifacts were recovered from the Nonda ZUS application. As for the FIXD application, partial vehicle-related artifacts were recovered. Finally, no vehicle-related artifact was recovered from the MB-Companion application.

Table 5.2 Comparison of vehicle related artifacts found from all Four testing Applications

	MB-Companion	ZUS-iOS	ZUS-Android	FIXD
VIN	NO	YES	YES	YES
Vehicle DTC	NO	YES	YES	NO

5.1.3 Research Question 3

The third research question was, “What different vehicle information can be recovered from the Mercedes and the Audi?” According to the results shown in Table 5.3, there was no difference between MB and Audi vehicles. The reason that only Audi’s VIN was recovered from the FIXD is that the mobile application only recorded the last connected vehicle. Since Audi was tested after the Mercedes vehicle, the records of the Mercedes testing vehicle were overwritten.

Table 5.3 Comparison of vehicle-related artifacts found from both devices for different vehicles

	FIXD	ZUS-iOS	ZUS-Android
Audi VIN	YES	NO	YES
MB VIN	NO	NO	YES
Audi DTC	NO	YES	YES
MB DTC	NO	YES	YES

5.1.4 Research Question 4

The fourth research question was, “What impacts from using different devices (Android and iOS) will occur when recovering the relevant artifact?” According to the results presented in Table 5.4, the artifacts recovered from the Android device provided a higher level of detail. For the Android device, most of the detailed artifacts were discovered from the realm database, which was also organized. However, most of the artifacts from the iOS device were recovered from the log files, which were not as detailed and organized as the artifacts recovered from a database.

Table 5.4 Comparison of artifacts found from the different mobile devices

	ZUS-iOS	ZUS-Android
Vehicle VIN	NO	YES
Vehicle ID	YES	YES
Device Info	PARTIAL	YES
User Info	PARTIAL	YES
User Password	NO	NO
Trip Info	PARTIAL	YES
DTC Info	PARTIAL	YES
Geofence Info	PARTIAL	YES

5.2 Recovered Artifacts Review

In this subsection, the characteristics of each mobile application are discussed regarding different operating systems (iOS and Android).

5.2.1 Mercedes Benz Companion – iOS

From the iOS advanced logical image created using the Cellebrite UFED 4PC, the artifacts recovered from the MB Companion mobile application are extremely limited. No vehicle-related artifacts and geolocation data were discovered. The extracted database file was either encrypted or damaged during the acquisition. Only two KTX files that contain snapshots of parking locations with only street names were recovered. However, those KTX files cannot guarantee the availability of artifacts like this all the time. KTX refers to Khronos Texture, which is an image file that is automatically generated by taking a screenshot of the application when the user switches from the currently used application to a different application or returns to the home screen.

5.2.2 Nonda ZUS – iOS

From the iOS advanced logic image created by the Cellebrite UFED, the artifacts recovered from the Nonda ZUS mobile application are quite abundant. Two types of log files were recovered under the package file path. In those log files, artifacts like the user account information, vehicle data, and trip information, including timestamp and GPS coordinates, can be recovered. Moreover, the created timestamp of the trip image can be found in the file metadata using the Cellebrite reader. The trip preview image with the start and endpoints can further be discovered in the image caches folder. A route can be recreated using the recovered artifacts. However, most of the artifacts were discovered in the log files in the format of plain text (TXT) and ordered by the timestamp. Everything was recorded in lines of word messages. Tens of thousands of lines of pure text

messages will cause many troubles in locating and noting certain information, which is needed for future digital forensic investigation.

5.2.3 Nonda ZUS – Android

From the Android advanced logic image created by the Cellebrite UFED, the artifacts recovered from the Nonda ZUS mobile application are very comprehensive and detailed. Numerous organized and detailed artifacts were recovered with the discovery of the realm database. The recovered realm database contains detailed user account data, vehicle information, nonda device information, DTC scanning records, trip data, and detailed geolocation information. Based on the recovered artifacts, an extremely accurate route can be recreated. Nevertheless, the forensic tool, Cellebrite Reader, is not able to view and examine the realm database. The problem can be solved with the correct version of MongoDB Realm Studio. This Realm Studio is not downward compatible, which means the newest version is not able to view and edit the realm database with an older schema.

5.2.4 FIXD – Android

From the Android advanced logic image created by the Cellebrite UFED, the artifacts recovered from the FIXD mobile application are relatively impoverished. For the user account information, only email address information was recovered in an XML file from the application package. From a different XML file, the vehicle identification number was recovered, but only for the last connected vehicle. The timestamp for activities like opening the application and connecting to the vehicle via the OBD II adaptor can also be discovered. However, no DTC scanning activity was recovered. Unlike the ZUS application, no database was discovered in the application package filesystem.

5.3 Suggestions and Recommendations to the Digital Forensic Investigators

According to the previous discussion and results, Nonda ZUS has the most forensically relevant artifacts recovered for both iOS and Android devices. Primarily, the log files are available for both operating systems, which is the main focus, particularly for the iOS version. However, a detailed note will be necessary when searching for valuable artifacts from the iOS log files. For the Android version, focusing on the realm database is recommended. Once successfully located and extracted the realm database, it is vital to use the correct version of MongoDB Realm Studio to access the database.

For other mobile applications like Mercedes Benz Companion and FIXD, the lack of forensically relevant artifacts makes them secondary objectives. Yet, there is still a possibility of recovering some artifacts as complementary evidence. For the iOS MB Companion application, there is still a chance of getting critical information, such as parking location, destination information, and vehicle status from the KTX file. For the FIXD application, which mobile device establishes a connection with a certain vehicle at what time might be critical in some cases as well.

5.4 Limitations During the Research

There were several issues and difficulties during the research. An only mobile digital forensic investigation was discussed in this study. Hardware forensics, vehicle forensics, and cloud forensics were not discussed.

5.4.1 Data Population

During the data population stage, the first iPhone X is jailbroken, which causes the Mercedes Benz Companion application to be unable to access because the application is not able to run on a modified system. A new iPhone X with unmodified iOS was adopted.

The MB Companion application was successfully installed on the Android device via Google Play. However, unlike the iOS version, which supports vehicles built after 2016, the Android version only supports vehicles built after 2017. Since the testing vehicle was built in 2016, the study of Companion on Android was canceled.

The iOS version FIXD application requires iOS version 15 and higher. However, no forensics tool was able to process the acquisition for iOS 15 at the time of the experiment. The study of FIXD on iOS was canceled.

5.4.2 Data Extraction

The first Android testing device was a Samsung A51. After finishing the data population, the Cellebrite UFED Physical Analyzer was not able to create an advanced logical file system image because the OEM unlock option was disabled by the carrier on the device. Rooting the system is impossible with this option disabled. A new device with a successfully rooted system was adopted.

There was only one forensic tool used during the data extraction, which was the Cellebrite UFED Physical Analyzer. Adopting different forensic tools could result in different outputs.

5.4.3 Data Analysis

The database found in the iOS Mercedes Benz Companion was not able to be processed by Cellebrite Reader, SQLiteSPY, and DB Browser for SQLite. Hence the conclusion—whether the database was encrypted or damaged—was made.

5.5 Future Work

This study is primarily focusing on Nonda ZUS on iOS and Android, Mercedes Benz Companion only on iOS, and FIXD only on Android. The expansion of the scope of the research can be considered. For instance, different mobile applications, newer operating systems, and different vehicles can be explored. This study only adopted one forensics tool Cellebrite UFED. More forensics tools can be applied in future mobile forensic research, like magnet AXIOM and XRY.

REFERENCES

- Alexakos, C., Katsini, C., Votis, K., Lalas, A., Tzovaras, D., & Serpanos, D. (2021). Enabling Digital Forensics Readiness for Internet of Vehicles. *Transportation Research Procedia*, 52, 339–346. <https://doi.org/10.1016/j.trpro.2021.01.040>
- Al-Sabaawi, A., & Foo, E. (2019). *A Comparison Study of Android Mobile Forensics for Retrieving Files System*. 19.
- Baig, Z. A., Szewczyk, P., Valli, C., Rabadia, P., Hannay, P., Chernyshev, M., Johnstone, M., Kerai, P., Ibrahim, A., Sansurooah, K., Syed, N., & Peacock, M. (2017). Future challenges for smart cities: Cyber-security and digital forensics. *Digital Investigation*, 22, 3–13. <https://doi.org/10.1016/j.diin.2017.06.015>
- Bays, J., & Karabiyik, U. (2019). *Forensic Analysis of Third Party Location Applications in Android and iOS*. 1–6.
- Bortles, W., McDonough, S., Smith, C., & Stogsdill, M. (2017). *An introduction to the forensic acquisition of passenger vehicle infotainment and telematics systems data (No. 2017-01-1437)*. SAE Technical Paper.
- Campbell, S., O'Mahony, N., Krpalcova, L., Riordan, D., Walsh, J., Murphy, A., & Ryan, C. (2018). Sensor Technology in Autonomous Vehicles: A review. *2018 29th Irish Signals and Systems Conference (ISSC)*, 1–4. <https://doi.org/10.1109/ISSC.2018.8585340>
- Caviglione, L., Wendzel, S., & Mazurczyk, W. (2017). The future of digital forensics: Challenges and the road ahead. *IEEE Security & Privacy*, 15(6), 12–17.

- Cebe, M., Erdin, E., Akkaya, K., Aksu, H., & Uluagac, S. (2018). Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles. *IEEE Communications Magazine*, 56(10), 50–57.
<https://doi.org/10.1109/MCOM.2018.1800137>
- Chang, H. H., & Cheon, S. H. (2019). The potential use of big vehicle GPS data for estimations of annual average daily traffic for unmeasured road segments. *Transportation*, 46(3), 1011–1032.
- Choo, K.-K. R., Gai, K., Chiaraviglio, L., & Yang, Q. (2021). A multidisciplinary approach to Internet of Things (IoT) cybersecurity and risk management. *Computers & Security*, 102, 102136. <https://doi.org/10.1016/j.cose.2020.102136>
- Ebbers, S., Ising, F., Saatjohann, C., & Schinzel, S. (2021). Grand Theft App: Digital Forensics of Vehicle Assistant Apps. *The 16th International Conference on Availability, Reliability and Security*, 1–6. <https://doi.org/10.1145/3465481.3465754>
- Feng, X., Dawam, E. S., & Amin, S. (2017). A New Digital Forensics Model of Smart City Automated Vehicles. *2017 IEEE International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCoM) and IEEE Smart Data (SmartData)*, 274–279. <https://doi.org/10.1109/IThings-GreenCom-CPSCoM-SmartData.2017.47>
- Feng, X., Dawam, E. S., & Li, D. (2019). *Autonomous Vehicles' Forensics in Smart Cities*. 1688–1694.
- FIXD - Vehicle Health Monitor—Apps on Google Play*. (2022).
https://play.google.com/store/apps/details?id=com.fixdapp.two&hl=en_US&gl=US
- FIXD Official Site*. (2021). FIXD Official Site. <https://www.fixd.com/>

- Fukami, A., Ghose, S., Luo, Y., Cai, Y., & Mutlu, O. (2017). Improving the reliability of chip-off forensic analysis of NAND flash memory devices. *Digital Investigation*, 20, S1–S11. <https://doi.org/10.1016/j.diin.2017.01.011>
- Gabler, H. C., Hinch, J. A., & Steiner, J. (2008). *Event Data Recorder. A Decade of Innovation*.
- Holt, T. J., Bossler, A. M., & Seigfried-Spellar, K. C. (2017). *Cybercrime and digital forensics: An introduction*. Routledge.
- Jacobs, D., Choo, K.-K. R., Kechadi, M.-T., & Le-Khac, N.-A. (2017). Volkswagen Car Entertainment System Forensics. *2017 IEEE Trustcom/BigDataSE/ICSS*, 699–705. <https://doi.org/10.1109/Trustcom/BigDataSE/ICSS.2017.302>
- Jayakumar, H., Raha, A., Kim, Y., Sutar, S., Lee, W. S., & Raghunathan, V. (2016). *Energy-efficient system design for IoT devices*. 298–301.
- Jiang, X., Lora, M., & Chattopadhyay, S. (2020). An Experimental Analysis of Security Vulnerabilities in Industrial IoT Devices. *ACM Transactions on Internet Technology*, 20(2), 1–24. <https://doi.org/10.1145/3379542>
- Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). Guide to integrating forensic techniques into incident response. *NIST Special Publication*, 10(14), 800–886.
- Khronos Texture*. (2021, August 18). FileInfo.Com. <https://fileinfo.com/extension/ctx>
- Kopencova, D., & Rak, R. (2020). Issues of Vehicle Digital Forensics. *2020 XII International Science-Technical Conference AUTOMOTIVE SAFETY*, 1–6. <https://doi.org/10.1109/AUTOMOTIVESAFETY47494.2020.9293516>
- KTX Overview*. (2022). The Khronos Group Inc. <https://www.khronos.org/ctx/>

- Lacroix, J., El-Khatib, K., & Akalu, R. (2016). Vehicular Digital Forensics: What Does My Vehicle Know About Me? *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, 59–66.
<https://doi.org/10.1145/2989275.2989282>
- Le-Khac, N.-A., Jacobs, D., Nijhoff, J., Bertens, K., & Choo, K.-K. R. (2020). Smart vehicle forensics: Challenges and case study. *Future Generation Computer Systems*, 109, 500–510. <https://doi.org/10.1016/j.future.2018.05.081>
- Li, S., Choo, K.-K. R., Sun, Q., Buchanan, W. J., & Cao, J. (2019). IoT Forensics: Amazon Echo as a Use Case. *IEEE Internet of Things Journal*, 6(4), 6487–6497.
<https://doi.org/10.1109/JIOT.2019.2906946>
- Losavio, M. M., Chow, K. P., Koltay, A., & James, J. (2018). The Internet of Things and the Smart City: Legal challenges with digital forensics, privacy, and security. *Security and Privacy*, 1(3), 1(3), e23.
- Lott, D. A., Raglin, A., & Metu, S. (2019). *On the use of Operations Research for Decision Making with Uncertainty for IoT devices in battlefield situations*. 266–297.
- MacDermott, A., Baker, T., & Shi, Q. (2018). Iot Forensics: Challenges for the Ioa Era. *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1–5. <https://doi.org/10.1109/NTMS.2018.8328748>
- Malekian, R., Moloisane, N. R., Nair, L., Maharaj, B. T., & Chude-Okonkwo, U. A. K. (2017). Design and Implementation of a Wireless OBD II Fleet Management System. *IEEE Sensors Journal*, 17(4), 1154–1164. <https://doi.org/10.1109/JSEN.2016.2631542>

- Mansor, H., Markantonakis, K., Akram, R. N., Mayes, K., & Gurulian, I. (2016). Log Your Car: The Non-invasive Vehicle Forensics. *2016 IEEE Trustcom/BigDataSE/ISPA*, 974–982. <https://doi.org/10.1109/TrustCom.2016.0164>
- Meneghello, F., Calore, M., Zucchetto, D., Polese, M., & Zanella, A. (2019). IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet of Things Journal*, 6(5), 8182–8201.
- Meola, A. (2020, March 10). How 5G & IoT technologies are driving the connected smart vehicle industry. *INSIDER*.
- Montasari, R., & Hill, R. (2019). *Next-generation digital forensics: Challenges and future paradigms*. 205–212.
- Morgul, E. F., Ozbay, Iyer, S., & Holguin-Veras, J. (2013). *Commercial Vehicle Travel Time Estimation in Urban Networks using GPS Data from Multiple Sources*. Transportation Research Board 92nd Annual Meeting (No. 13-4439).
- Mukhopadhyay, D., Gupta, M., Attar, T., Chavan, P., & Patel, V. (2018). An Attempt to Develop an IOT Based Vehicle Security System. *2018 IEEE International Symposium on Smart Electronic Systems (ISES) (Formerly INiS)*, 195–198. <https://doi.org/10.1109/iSES.2018.00050>
- Mylonas, A., Meletiadis, V., Tsoumas, B., Mitrou, L., & Gritzalis, D. (2012). Smartphone Forensics: A Proactive Investigation Scheme for Evidence Acquisition. *Information Security and Privacy Research*, 376, 249–260. https://doi.org/10.1007/978-3-642-30436-1_21
- Neeb, J. E. (2004). *Joint test action group (JTAG) tester, such as to test integrated circuits in parallel* (U.S. Patent and Trademark Office Patent No. 6766486).

- Prastya, S. E., Riadi, I., & Luthfi, A. (2017). Forensic Analysis of Unmanned Aerial Vehicle to Obtain GPS Log Data as Digital Evidence. *IJCSIS*, 15(3), 7.
- Ruengittinun, S., Paisalwongcharoen, J., & Watcharajindasakul, C. (2017). *IoT solution for bad habit of car security*. 1–4.
- Salamh, F. E., Mirza, M. M., Hutchinson, S., Yoon, Y. H., & Karabiyik, U. (2021). What's on the Horizon? An In-Depth Forensic Analysis of Android and iOS Applications. *IEEE Access*, 9, 99421–99454. <https://doi.org/10.1109/ACCESS.2021.3095562>
- Saufi, N. N. C., Razak, N. S. M., & Mansor, H. (2019). FoRent: Vehicle forensics for car rental system. *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy - ICCSP '19*, 153–157. <https://doi.org/10.1145/3309074.3309101>
- Sehgal, V. K., Mehrotra, S., & Marwah, H. (2016). Car security using Internet of Things. *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 1–5. <https://doi.org/10.1109/ICPEICES.2016.7853207>
- Shah, Y., & Sengupta, S. (2020). *A survey on Classification of Cyber-attacks on IoT and IIoT devices*. 0406–0413.
- Sikder, A. K., Petracca, G., Aksu, H., Jaeger, T., & Uluagac, A. S. (2018). A Survey on Sensor-based Threats to Internet-of-Things (IoT) Devices and Applications. *ArXiv:1802.02041 [Cs]*. <http://arxiv.org/abs/1802.02041>
- Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., & Markakis, E. K. (2020). A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys & Tutorials*, 22(2), 1191–1221. <https://doi.org/10.1109/COMST.2019.2962586>

- Sun, P., Li, J., Bhuiyan, M. Z. A., Wang, L., & Li, B. (2019). Modeling and clustering attacker activities in IoT through machine learning techniques. *Information Sciences*, 479, 456–471.
- Suri, N., Tortonesi, M., Michaelis, J., Budulas, P., Benincasa, G., Russell, S., Stefanelli, C., & Winkler, R. (2016). *Analyzing the applicability of internet of things to the battlefield environment*. 1–8.
- Tang, C., Wei, X., Zhu, C., Wang, Y., & Jia, W. (2020). Mobile vehicles as fog nodes for latency optimization in smart cities. *IEEE Transactions on Vehicular Technology*, 69(9), 9364–9375.
- White, J., Thompson, C., Turner, H., Dougherty, B., & Schmidt, D. C. (2011). WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones. *Mobile Networks and Applications*, 16(3), 285–303. <https://doi.org/10.1007/s11036-011-0304-8>
- Wurm, J., Hoang, K., Arias, O., Sadeghi, A. R., & Jin, Y. (2016). *Security analysis on consumer and industrial IoT devices*. 519–524.
- Zunic, E., Hindija, H., Besirevic, A., Hodzic, K., & Delalic, S. (2018). Improving Performance of Vehicle Routing Algorithms using GPS Data. *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, 1–4.
<https://doi.org/10.1109/NEUREL.2018.8586982>