Spring 2018

Security attacks on RECKLESS-APPs: "Remote car keyless applications" for new semi autonomous vehicles

Mohamed El-Tawab

Security Attacks on RECKLESS-APPs: ”Remote Car keyless Applications” for new semi Autonomous Vehicles

Mohamed El-Tawab

A thesis submitted to the Graduate Faculty of

JAMES MADISON UNIVERSITY

In

Partial Fulfillment of the Requirements

for the degree of

Master of Science

Department of Computer Science

May 2018

FACULTY COMMITTEE:

Committee Chair: M. Hossain Heydari

Committee Members/Readers:

Florian Buchholz

Ahmad Salman

Dedication

This work is dedicated to my parents. Without your support, I would not be here standing where I am as a young gentleman. I will forever be thankful for you!

I love you both.

Mohamed

Acknowledgments

First, I would like to thank my advisor, Dr. Mohammed Heydari, for his support throughout the completion of my thesis. Second, I would like to thank Dr. Florian Buchholz and Dr. Ahmad Salman for serving on my thesis committee. Third, I would like to thank my brother Dr. Samy for his support throughout my two years at James Madison University.

Table of Contents

Dedication ii

Acknowledgments iii

List of Tables vii

List of Figures ix

Abstract x

1 Introduction 1

1.1 Motivations 2

1.2 Objectives and Goals 2

1.3 Contributions 2

2 Background and Related Work 4

2.1 Intelligent Transportation Systems: Connected and Autonomous Vehicles 4

2.2 Types of Communications in Intelligent Transportation Systems 4

2.2.1 Vehicle-to-Vehicle - V2V 5

2.2.2 Vehicle-to-Infrastructure - V2I 5

2.2.3 Vehicle-to-Pedestrian - V2P 5

2.3 Features of Applications in Intelligent Transportation 6

2.4 A key/keyless System for Vehicles 6

2.4.1 Smartphones Applications 7

2.4.2 Cyber Attacks on Keyless Remote Entry/Ignition Systems 9

3 Analysis and Design of RECKLESS-APPs 11

3.1 Analysis of RECKLESS-APPS in the Market 11

3.1.1 Jeep-Uconnect 11

3.1.2 BMW Connected 12

3.2 Comparison between two of smartphone applications in the appstore 13

3.3 Design of RECKLESS-APP simulation 14
3.3.1 iPhone iOS Application: Design and Mockups 15
3.3.2 Backend Server acts as the vehicle 16

4 Simulation of RECKLESS-Apps using an iOS Approach 17

4.1 Overview of Hardware and Software 17
4.1.1 Mobile Development 17
4.1.2 The Vehicle Server 18
4.1.3 Database 18
4.2 Methodology of Communication 18
4.2.1 System Architecture Overview 18
4.2.2 Application Flow and Data Flow Diagram 19
4.3 Mobile Application 20
4.3.1 Welcome and Sign-Up Pages 21
4.3.2 Home and User Information Side Pages 22
4.3.3 Location and Lock/Unlock Pages 23
4.4 Database 23

5 Security Attacks on RECKLESS-APPs 26

5.1 Cyber Attacks on Vehicles and Applications 26
5.1.1 Man-in-the-Middle(MITM) Attack 26
5.1.2 Social Engineering Attack 27
5.1.3 Application-oriented Attack 28
5.1.4 Denial of Service Attack 28
5.1.5 Other Attacks 28
5.2 Attacking RECKLESS-Apps 29
5.2.1 Applying Evil Twin Attack 29
5.2.2 Applying Man in the Middle Attack 32
5.2.3 Summary of Attacks 33
5.3 Mitigation solutions and recommendations to protect RECKLESS-Apps from Cyber attacks 33
5.3.1 Evil Twin Attack Mitigation 33
5.3.2 Man in the Middle Mitigation 34
5.4 Summary of Attacks and Mitigation 35

6 Conclusion and Future Work 36

6.1 Conclusion . . . 36

6.2 Future Work . . . 37

Bibliography 38

List of Tables

3.1 Comparison between Jeep-Uconnect and BMW-Connected 14

5.1 Summary of attacks and mitigation 35

List of Figures

2.1 V2X communication . . . 4

2.2 Combined V2I and V2V communication . . . 5

2.3 Keyless System for vehicles . . . 7

2.4 Smartphone-based accident detection system . . . 8

2.5 Smartphone App Keyless systems . . . 9

2.6 How these applications communicate with the vehicles . . . 9

3.1 Jeep-UConnect . . . 12

3.2 BMW-Connected . . . 13

3.3 Architecture Diagram . . . 15

3.4 App-Registration . . . 15

3.5 App-Functionality . . . 16

4.1 System Architecture . . . 19

4.2 Data Flow Diagram . . . 20

4.3 Application Flow Diagram . . . 21

4.4 Welcome and Sign-Up pages . . . 22

4.5 Home and User Information Side Pages . . . 22

4.6 Location and Lock/Unlock pages . . . 23

4.7 Users table . . . 24

4.8 Cars table . . . 24

4.9 Users-Cars table . . . 24

4.10 Entity Relationship Diagram . . . 25

5.1 Man in the middle attack . . . 27

5.2 MITM Attack on Different Layers of OSI Model and Types of Cellular Networks . . . 27

5.3 DOS Attack . . . 28

5.4 Connection between the mobile application and the vehicle . . . 29

5.5 Starbucks Free WiFi . . . 30

5.6 The victim is using the attacker's WiFi 30

5.7 Mobile Data Consumption Attack 31

5.8 Denial of Service Attack 32

5.9 Summary of Attacks 33

Abstract

Rapid technological advancements of vehicle manufacturing and the modern wireless technology opens the door for several new Intelligent Transportation applications. Remote Keyless system in vehicles is considered one of the famous applications that has been developed recently, which is susceptible to many cyberattacks. Remote Keyless applications on smartphones were developed in the past few years to perform the functionality of keyless fob and are expected to replace the physical keyless fobs in the next few years, which can open the door to many cyberattacks. In this research, we implemented a simulation that represents the REmote Car KeyLESS Applications (RECKLESS-apps) on the smartphones. In this thesis we demonstrate the types of cyberattacks these applications could face. Our research serves as a proof that remote car keyless applications that would replace the physical keys could be vulnerable to various malicious cyberattacks. We study these attacks and provide remedies, cyber-hygiene and best practices to remedy some of these attacks.

Chapter 1

Introduction

Rapid technological advancements of vehicle manufacturing and the modern wireless technology opens the door for several new Intelligent Transportation applications. Remote Keyless system in vehicles is considered one of the famous applications that has been developed recently, which is susceptible to many cyberattacks. Remote Keyless applications on smartphones were developed in the past few years to perform the functionality of keyless fob and are expected to replace the physical keyless fobs in the next few years, which can open the door to many cyberattacks. In this research, we implemented a simulation that represents the REmote Car KeyLESS Applications (RECKLESS-apps) on the smartphones. In this thesis we demonstrate the types of cyberattacks these applications could face. Our research serves as a proof that remote car keyless applications that would replace the physical keys could be vulnerable to various malicious cyberattacks. We study these attacks and provide remedies and cyber-hygiene and best practices to remedy some of these attacks.

In the last few years, Remote Car Keyless applications were released on the smartphones which perform the functionality of the physical key fob. Moreover, companies (e.g., BMW) are expecting to entirely replace the key fobs with smartphones applications [1]. The idea is not very popular at the time and is under construction. These applications open the door to many malicious cyberattacks. In this research, we developed a Remote Car Keyless application on a smartphone, by developing an iOS application. We simulated cyberattacks, such as Evil Twin Attack and Man in the Middle Attack, and discussed how these attacks could be mitigated.

In Chapter 2, we provide general background information and related work, to support different components of our thesis as well as highlighting the overall history of security in Intelligent Transportation Systems (ITS). We discuss the history of the remote keys used for vehicles. In Chapter 3, we show the analysis and design of the Remote Car keyless Applications on smartphones. We highlight several existing applications in the app store and describe the architecture diagram. Also, we emphasis the simulation design of the iOS application we developed for our system and the backend server. In Chapter 4, we show the actual implementation of the iOS mobile application, using Objective-C, and the sequence diagram of how to use the app and its communication with the backend server. In Chapter 5, we study the security attacks that can occur on vehicles and the mobile application, such as Traffic Analysis attack, Social Engineering attack, Denial of Service attack, etc. Then, we show types of attacks performed on our system (Evil Twin Attack and Man in the Middle attack). Finally, we analyze some of these cyberattack vectors and provides mitigation techniques. In Chapter 6, we discuss the conclusion of our work and show the future work that can be applied to our system.

1.1 Motivations

The car thefts that have happened and recorded recently is the main motivation for our work. Thieves were able to steal a Mercedes Bens car without using a key in just seconds, in one of the first relay crimes caught on camera [2]. All this made us think about the current smartphone applications on the app store and how these applications are vulnerable to various malicious attacks.

These applications currently have less than average customer reviews, which might be an indication of how unsecured these applications currently are. It is time for researchers to focus on this hot topic by studying the vulnerability of these applications to various cyber-attacks.

1.2 Objectives and Goals

There are four main objectives/goals behind our research

- Explore current research done on the topic of *Smart Keys* for vehicles,
- Show how Remote Car Keyless applications work by implementing a simulated app,
- Perform cyberattacks on the Remote Car Keyless applications,
- Discuss what kind of mitigation techniques can be applied to improve the security of these apps.

1.3 Contributions

In this thesis we study using smartphones as smart car keys, to access/control vehicles remotely from any place using different types of communication. We implemented a simulation of how Remote Car

Keyless applications work. Also, we study the communication that happens between the smartphone application and the backend server, which acts as the victims car. We perform two cyberattacks (Evil Twin Attack and Man in the Middle attack), and then we discuss what kind of mitigation should be applied to improve vulnerable to Evil Twin and Man in the Middle attacks.

Chapter 2

Background and Related Work

In this chapter, we provide general background information and related work to support different components of our thesis as well as highlight the overall history of security in Intelligent Transportation Systems (ITS) research. We highlight various ITS applications and provide examples of related transportation systems. We discuss the history of car key used for vehicles. In addition, we highlight the effect of wireless communication on the remote keyless entry in vehicles.

2.1 Intelligent Transportation Systems: Connected and Autonomous Vehicles

An Intelligent Transportation System (ITS) is a system that improves traffic safety, by notifying the driver about the current situation to take the proper action [3]. In Intelligent Transportation Systems, vehicles can communicate with different types of objects (e.g., communication with other vehicles (Vehicle-to-Vehicle (V2V)), communicate with roadside units (RSU) (Vehicles-to-Infrastructure (V2I)). Figure 2.1 shows different types of communication that can occur in the ITS.

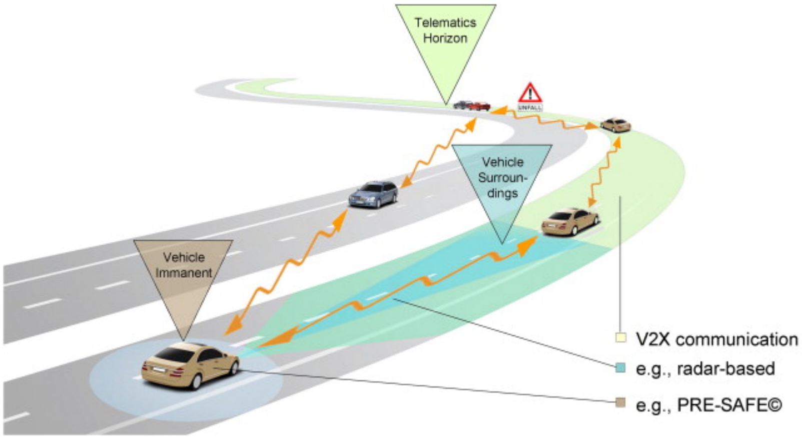


Figure 2.1: V2X communication [4]

2.2 Types of Communications in Intelligent Transportation Systems

In this section, We introduce in more details different types of ITS communications. These types of communications usually adopt standard Dedicated Short Range Communications (DSRC) for fast communications which are shortrange wireless communication channels specially designed for automotive use and a corresponding set of protocols and standards (IEEE 802.11p [5, 6]). The transmission range of DSRC is 300 meters (new DSRC assume 1000 meters). DSRC for intelligent transportation systems operates in the 5.9 GHz and (U.S.) or 5.8 GHz band (Japan, Europe). In all these types of communications, safe communication channels are needed for the wireless transmission.

2.2.1 Vehicle-to-Vehicle - V2V

Vehicle-to-Vehicle (V2V) communication is the wireless transfer of data between vehicles. Information such as speed, location, direction, acceleration and deceleration are different types of data that can be send/receive between different vehicles.

2.2.2 Vehicle-to-Infrastructure - V2I

Vehicle-to-Infrastructure (V2I) communication is the wireless transfer of data between vehicles and infrastructure of the road. The wireless communication occurs between the on-board unit (OBU) and the road side units (RSU). Several applications are using these types of communication such as Traffic Control for emergency vehicles, and Traffic Sign Recognition (TSR) [7]. Figure 2.2 shows an example of V2I and V2V communication.

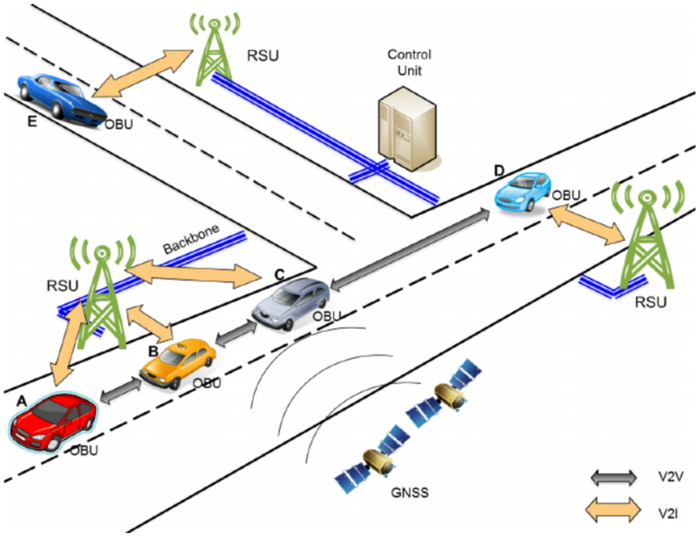


Figure 2.2: Combined V2I and V2V communication [8].

2.2.3 Vehicle-to-Pedestrian - V2P

Vehicle-to-Pedestrian (V2P) communication is the wireless transfer of data between vehicles and pedestrians, it takes advantage of hardware included in smartphones of pedestrians, such as Global

Positioning System(GPS) sensor and accelerometers, to make people on the street "visible" to the vehicles [9]. Several researchers have been studying these types of applications [10].

2.3 Features of Applications in Intelligent Transportation

In general, Intelligent Transportation applications have four different fundamental demands: scalability, availability, context-awareness, and security and privacy [11, 12].

1. *Scalability:* Because of the number of vehicles that could be incorporated into Vehicular Networks, Vehicular Networks may become the largest ad hoc network in the history. Undoubtedly, scalability will be a critical factor. The use of hybrid architecture, together with in-network aggregation techniques and Peer-to-Peer (P2P) technologies, make information exchange more scalable.

2. *Availability:* Due to the real-time interaction between vehicular networks and the physical world, availability is an important factor in system design. This may have a major impact on the safety and efficiency of future highway systems. The architecture should be robust enough to withstand unexpected system failures or deliberate attacks.

3. *Context-Awareness:* As a cyber-physical system, applications in intelligent transportation collect information from the physical world and may conversely impact the physical world.

4. *Security and Privacy:* There is a recent trend of making vehicular on-board computer systems inter-connectable to other systems. The Ford Sync, for example, connects the vehicles entertainment system to the drivers cell phone via blue-tooth technology. In the future, vehicular on-board computers could even be open to software developers. These trends may have serious implications for security and privacy due to the cyber-physical nature of these applications. Governments and consumers will have very high expectations of safety and security.

2.4 A key/keyless System for Vehicles

The key for the vehicle has been modified for years to help drivers to unlock/lock the vehicle, adding different functionalities. The keyless entry system was introduced in 1980 by Ford, Mercury and Lincoln. Later, several other companies introduced keyless remote key in their vehicles. The early stage of the Keyless remotes contain a short-range radio transmitter, and must be within a certain range of 520 meters, of the car to operate [13]. The idea is very simple, when a driver pushes the button in the remote, it sends a coded signal by radio waves to a receiver unit in the car, which locks or unlocks the door. Most remote keyless entry (RKE)s operate at a frequency of 315 MHz for North America-made cars and at 433.92 MHz for European, Japanese and Asian cars [13]. After a number of incidents in the mid-1990$'s$, vehicle manufactories implemented encryption as well as rotating entry codes to prevent car thieves from intercepting and spoofing the signal. In early 2000, the idea of remote keyless ignition system was introduced that can in addition of locking and unlocking the door, can start the engine. Figure 2.3 shows different types of keyless systems.

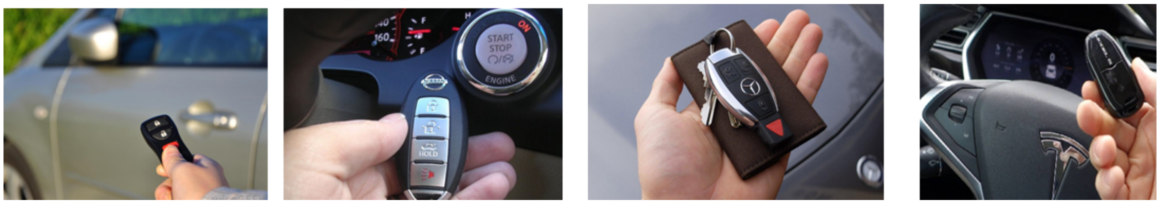


Figure 2.3: Keyless System for vehicles[14][15][16][17]

Nowadays, most of the new vehicles are using a smart key. Vehicle Manufactories are looking for new ways to use smartphone capabilities. The fact that almost everyone carries a smartphone and there are mobile app which allows car owners to perform tasks on their cars, make fashioned keys less relevant [1]. Security and privacy of the users of these apps are very critical. Several researchers have pointed to the importance of studying the privacy and security of these new systems but until now, as far as we researched, no one did a complete study of the possible attacks on these new systems.

2.4.1 Smartphones Applications

There are a large number of applications for smartphones that have a deep impact in ITS, applications that improve and help the transportation systems. Smartphone applications in Intelligent Transportation include data collection and management [18], tracking vehicle and anti-theft System [19], applications that help drivers reduce the fuel consumption of their vehicles by evaluating driving efficient patterns[20], applications for smart parking [21, 22] and applications for keyless smart systems [23]. In addition, Driving Style Recognition is one of the applications that used to help in vehicle safety systems, they detect, recognize and record aggressive actions without external processing in order to increase awareness and promote driver safety using Dynamic Time Warping and smartphone based sensor-fusion (accelerometer, GPS, video ...etc)[24].

Traffic Accident Detection and Notifications with smartphones is another example where it uses an approach to reduce delay between accident occurrence and first responder dispatch, automatically detects traffic accidents using accelerometers and acoustic data and immediately notifies a central emergency dispatch server after an accident[25]. Figure 2.4 shows an example of a Smartphone-based accident detection system.

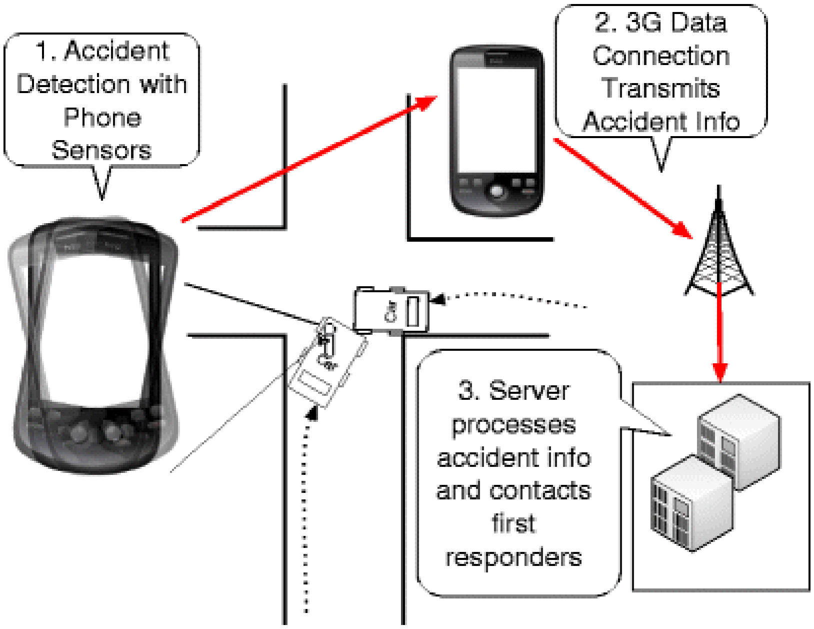

1. Accident Detection with Phone Sensors
2. 3G Data Connection Transmits Accident Info
3. Server processes accident info and contacts first responders
Car
Car

Figure 2.4: Smartphone-based accident detection system[25].

A Driver profiling systems is another application of using smartphones and mobile devices to identify risky driving maneuvers and to improve driver efficiency. It can be useful in fleet management, insurance premium adjustment, fuel consumption optimization or CO2 emission reduction[26]. Several car manufactures started to have apps that can control the vehicles, performs functions such as: Lock/unlock, start the vehicle, open the windows, flash lights, honk horn, and Open windows and roof. Figure 2.5 shows different examples of smart phone applications on app store and google play for starting and controlling some functionalities of the vehicle.

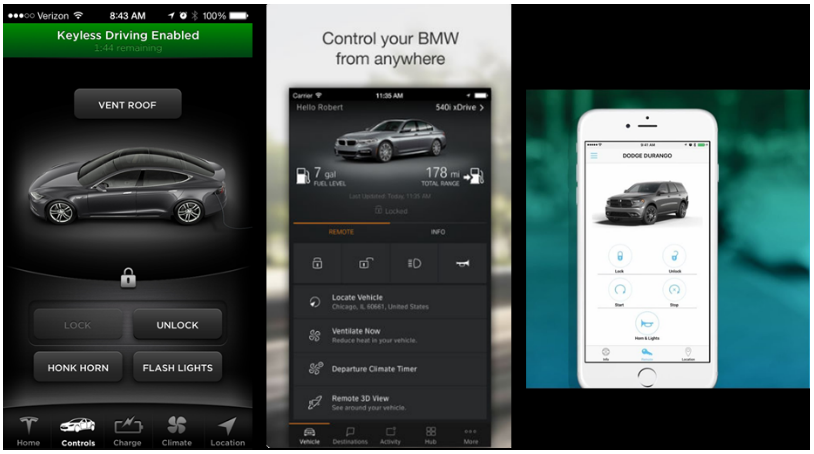


Figure 2.5: Smartphone App Keyless systems[27][28][29].

Despite the fact that developers and Intelligent Transportation experts are eager to integrate smartphones with Intelligent Transportation System applications, the security of these systems must be checked to prevent fatal consequences. Cyber attacks on these system are important to identify to protect drivers, vehicles and the privacy of all involved.

2.4.2 Cyber Attacks on Keyless Remote Entry/Ignition Systems

It is important to discuss the cyber-security implications of ITS and the potential cyber-attacks specific to these types of applications. We will summarize the threats on ITS and cooperative automated vehicles. Communication uses cellular network to connect with the smart phone, Bluetooth is used to connect to the vehicle for application such as audio, video applications in the vehicle. Figure 2.6 shows the communication between the vehicle and driver's smartphone or computer.

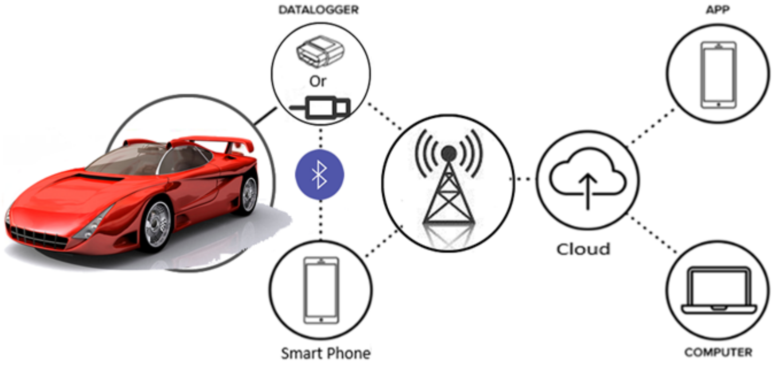


Figure 2.6: How these applications communicate with the vehicles[30]

The Remote Keyless Ignition (RKI) does not provide much security by default. In October 2014, some insurers in the United Kingdom (UK) would not insure certain vehicles with keyless ignition without buying security devices [31]. Several cases were reported where criminals were able to open cars by tricking vehicles to think that their keyless entry fobs were close enough, although they were far away [32]. In 2015, it was reported that an inexpensive electronic device about the size of a wallet was built that could be placed near a locked vehicle to capture the keyless entry code that is used to unlock the vehicle. In 2017, a group of researchers pulled off the relay hack with a pair of gadgets they built for just $22 .

Smartphone applications for Remote Keyless started in 2011 by Jeep and other manufacturers. There are growing number of smartphone application users who wishes to use their smartphone to control their vehicles. Security has not yet been studied sufficiently to ensure that these applications are not vulnerable to security attacks. Attacks such as: Denial of service attack, sniffing attack, man in the middle attack (MiTM), or other attacks must be studied.

In this thesis, we study various attacks on smartphone applications. We designed and developed a smartphone application for the keyless ignition, using iOS for testing. We analyzed various attacks that can be performed on these types of smartphone applications. Our results show that a more powerful security must be added to these types of systems.

Chapter 3

Analysis and Design of RECKLESS-APPs

In this chapter, we show the analysis and design of the Remote Car keyless Applications for smartphones. We highlight several existing applications in the market and we describe their architecture diagram. Also, we present the simulation design of the iOS application used for our system. Finally, we discuss the design of the backend server.

3.1 Analysis of RECKLESS-APPS in the Market

Nowadays, every car manufacture has it's own application on the smartphone stores, but a very limited number of them have Keyless Remote App embedded in the application that allows the functionality of the physical keyless system. In order to design our simulation app we will show and discuss few of the applications that currently exist to see their basic functionality. Despite the fact that smartphone applications exists on both the apple(iOS) and google play stores(Android), we focus on the applications on the appstore. In the following sections, we show the functionality of two different applications from two different car manufactures: Jeep and BMW.

3.1.1 Jeep-Uconnect

In the Jeep manufactory, in order to be able to use the smartphone application, you need to register your Uconnect System and your Jeep vehicle must be properly equipped with compatible Uconnect system. Uconnect features started in 2013. The Jeep Uconnect Application is simple in User Interface (UI) design. figure 3.1 shows the functionality of the Uconnect app.

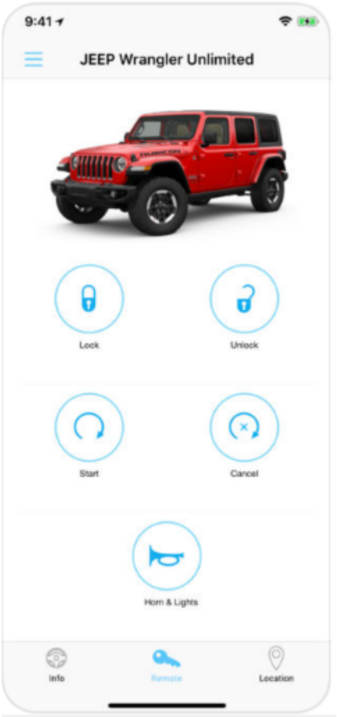


Figure 3.1: Jeep-UConnect[29]

In Uconnect application, users can perform the following functions [29]:

- Lock or unlock your car from virtually anywhere.
- Start your car and cancel start remotely.
- Flash the lights and sound the horn to help find your car.
- Locate your car on a map using Vehicle Finder.

3.1.2 BMW Connected

"BMW Connnected" is another example of a smartphone iOS application that requires an active BMW ConnectedDrive subscriptions. The vehicle must also be properly equipped with compatible BMW ConnectedDrive system (cellular network) in order to use this application. The app is optimized for vehicles from 2014 and newer. Elegance and excessive functions are very obvious in this application with sophisticated User Interface (UI) design. Figure 3.2 shows the BMW-connected smartphone application.

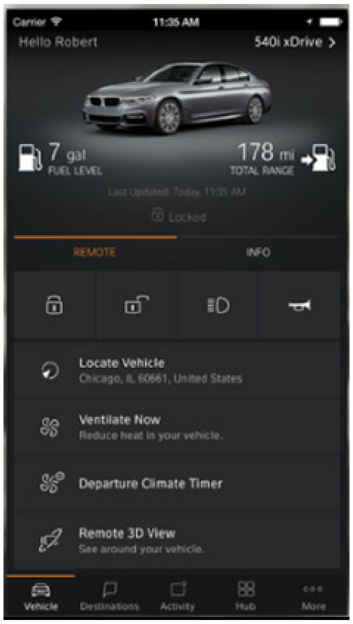


Figure 3.2: BMW-Connected[28]

In BMW Connected smartphone app, users can perform the following functions [28]:

- Lock or unlock your BMW and activate the climate control, anytime and anywhere.
- Send destinations from the app to your in-car navigation system.
- Find your car.
- Schedule trips, get alerted when to leave (based on traffic), and find parking spot nearby.
- Track your driving efficiency and manage battery charge.
- Access Roadside Assistance, schedule service appointments, and connect via Amazon Alexa or Google Assistant.

Both applications from Jeep and BMW have some common functionalities and some differences.

3.2 Comparison between two of smartphone applications in the appstore

After testing both applications (Jeep-Uconnect, BMW-Connected), we summarize their comparison in the table below:

Table 3.1: Comparison between Jeep-Uconnect and BMW-Connected

Remote Car keyless Applications		
Characteristic	Jeep-Uconnect	BMW-Connected
Free-App	✓	✓
Support Apple-Watch	✓	✓
Free-to-Register	✗	✓
Require paid Subscription	✓	✓
Require Cellular Network	✓	✓
Good Apple Store Reviews	✗	✗

As mentioned in the above table, both apps Jeep-Uconnect and BMW-Connected have less than average application reviews in the app-store. Jeep-Uconnect has a 2 out of 5 while BMW-Connected has a 2.5 out of 5 in March 2018. The non-satisfactory rating and the fact that these applications are relatively new were the reasons for us to investigate how these applications work and their vulnerability to cyber attacks [33].

3.3 Design of RECKLESS-APP simulation

As mentioned in the two previous sections, the Remote Car keyless Applications are considered relatively new and there are very few cars with subscribed cellular network. We designed an iOS smartphone application that simulates the Remote car keyless applications currently used in stores. Also, we replace the vehicle with a computer (server machine) that represents the computer embedded in the vehicle. The overall design of the system consists of mainly two parts, the first part is the smartphone application built on the iPhone plateform (iOS), the second part is the car manufacturer's service and the car system itself. We decided to use Linux Ubuntu server to act as our car. In the middle, we depend on the cellular network for connection to simulate the real connection between the two parts. Figure 3.3 shows that architecture diagram of our testing system.

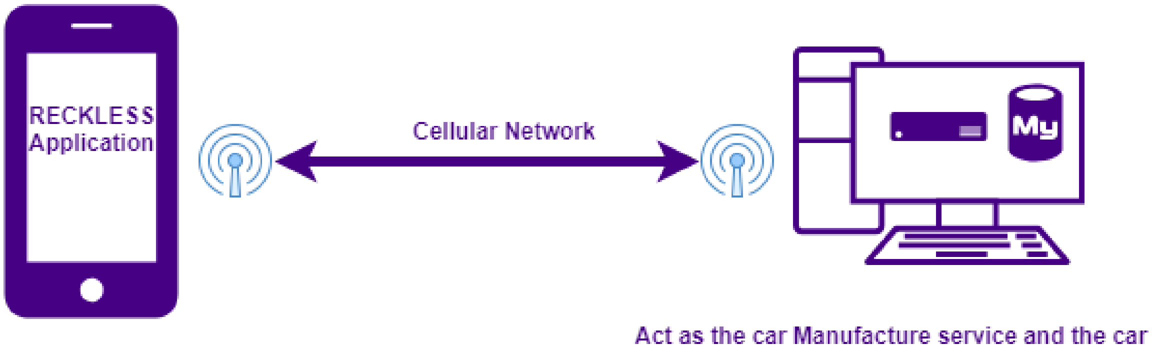


Figure 3.3: Architecture Diagram

3.3.1 iPhone iOS Application: Design and Mockups

We kept the application as simple as possible. It includes two main steps, the first one is the registration of a new account by providing the required information from the user about his/her car, the user can later use the created credentials to login to the system. figure 3.4 shows the registration page in the app, that allows new users to register to our system.

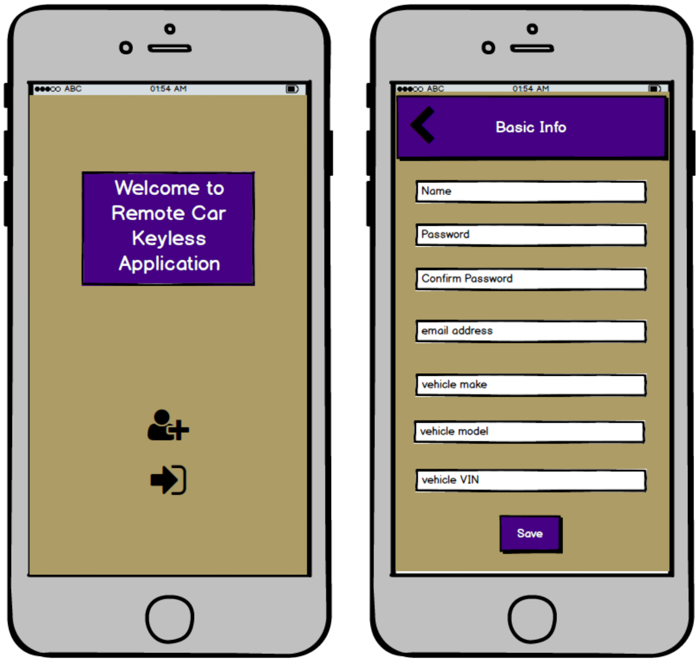


Figure 3.4: App-Registration

The second step is the functionality that the app could provide, includes locking/unlocking the car, start the engine, locating the car, honk horn and provide more info about the car service. Figure 3.5 shows the app-functionality, more details in the chapter 4.

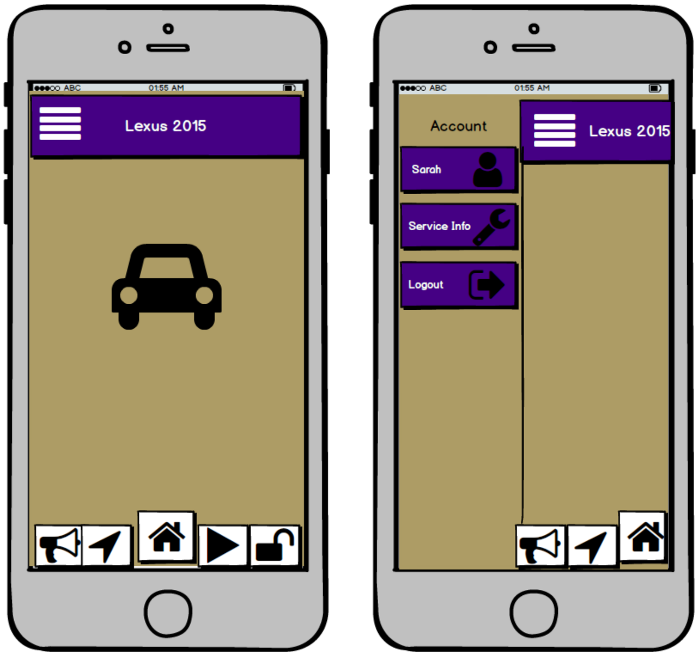


Figure 3.5: App-Functionality

For the implementation, we used Objective-C for the iOS development and we use Software Development Kit (SDKs) such as Apple MapKit [34]. Apples Objective-C using Xcode was chosen to create the iPhone application. Objective-C has the advantage of learning and developing a mobile app easier than some more advanced languages. Its main functionalities are for creating the user interface of the app, handling logic within the app, making API calls to get data from the database, and parsing through the returned data to be displayed via the user interface.

3.3.2 Backend Server acts as the vehicle

In order to complete our system, we used a Linux Ubuntu Host in the lab to act as the vehicle manufacturer's service and the vehicle. We created a RESTful Application Programming Interface (API) using Java that handles the request/response in JavaScript object notation (JSON), formatted for deployment onto a Tomcat application server. It connects to our database using Java Database Connectivity (JDBC).

Chapter 4

Simulation of RECKLESS-Apps using an iOS Approach

As we mentioned in Chapter 3, we created an iOS application as to be able to simulate how the Remote KeyLess Applications work. A user ”or a driver” simply uses our simulated RECKLESS-Application to create a new account providing his/her required information. After registration, each user can perform multiple functionalities that most of the current Remote Keyless Applications on the store perform. In order to complete our system, we assumed a server machine that acts as the vehicle manufacture service and the vehicle system. The iOS RECKLESS-Application communicates with the localhost server using RESTful JavaScript object notation (JSON) Application Programming Interface (API). Moreover, a database is created on the localhost to save/retrieve the user information. In this chapter, we discuss in details the hardware and software involved in our system, system architecture, database design and finally the application flow.

4.1 Overview of Hardware and Software

In this section, we discuss the mobile application functionality, the server details, and the database design.

4.1.1 Mobile Development

The iOS mobile application was developed using Apple's Objective-C language. The Objective-C language was chosen for a variety of reasons. First and foremost, it’s an object-oriented language. The kind of functionality that’s packaged in the Cocoa frameworks can only be delivered through object-oriented techniques. Second, because Objective-C is an extension of the standard ANSI C, existing C programs can be adapted to use the software frameworks without losing any of the work that went into their original development. Because Objective-C incorporates C, you get all the benefits of C when working within Objective-C [35]. The mobile app main functionalities are creating the user interface of the app, handling logic within the app, making API calls to get data from the host database, and parsing through the returned data to be displayed via the user interface. The API calls being made by Objective-C are hosted on the same environment as the database.

4.1.2 The Vehicle Server

The vehicle server plays the role of a vehicle by responding to different messages coming from our simulated mobile application. In addition, it provides the wireless communication capabilities that a vehicle may have such as Bluetooth, WiFi and Cellular Network communication [36].

4.1.3 Database

In order to keep track of the information needed for our simulation such as driver information, vehicle information, and functionalities, a MySQL database has been created and hosted on the server. The database contains information such as a "Users" entity and "Cars" entity. This database should be constantly updated to ensure the mobile application is displaying real-time information, when users check the applications at any given time as a part of the simulation. The MySQL database was created on an Ubuntu Server using a PHPMyAdmin GUI. The database, titled "RECKLESS", is comprised of three entities including, "users" , "cars" and "users_cars" storing necessary attributes. The "users" table will hold the users information (userId, name, hashed_password and email). The "cars" table will hold the cars information (carId, model, make, year and picture) and the "users_cars" tables will hold specific user's car information such as (userId, carId, VIN_number, service_Info, car_latitude, car_longitude, lock_status and engine_status). The database will be constantly updated to ensure users are getting accurate information when they check the mobile application at any given moment.

4.2 Methodology of Communication

4.2.1 System Architecture Overview

Before going into the details of the application, it will first help to step back and take a look at the simulation as a whole. The overarching goal of the simulation is to provide an application that allows users to check specific vehicle information and perform some functionalities to simulate the Remote Keyless Applications. There are two components for this system: the backend component which is the vehicle server host simulating the vehicle manufacturer's service and vehicle system. The front end mobile app itself is simply the iPhone application. As users navigate through the application, the app pulls responses with information from the database, almost at real time. The application also may request to update some information such as Lock/Unlock status of the vehicle. The application communicates with the backend with a cellular network to simulate the communication in the real Remote Keyless Applications. The backend is simply a Linux server that is running a Tomcat application server to handle the request/response to update/retrieve information in JSON formate. Figure 4.1 shows the system architecture.

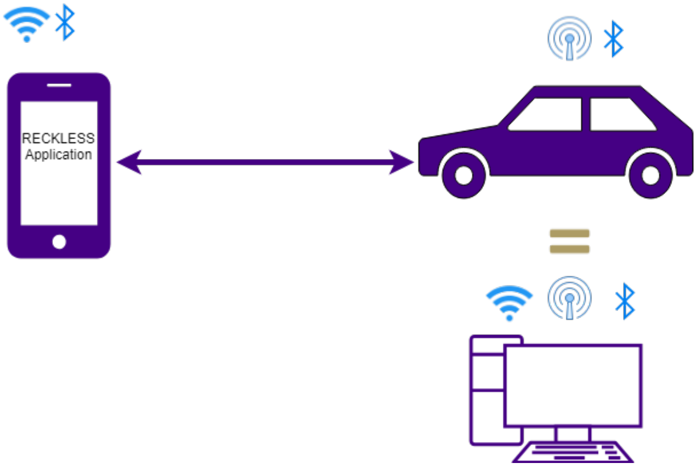


Figure 4.1: System Architecture

4.2.2 Application Flow and Data Flow Diagram

The user starts the app and the welcome screen pops up with Signin/Signup options. The user can sign up for a new account where he/she can specify different attributes for his/her information(name, username, password, email) and car information (VIN number , car model, car make and year), as soon as the password on the frontend matches, a POST request will be sent to the backend server with the mentioned information, the user information (name, username, hashed password, email) will be stored in the "users" table with a new (user_id), a new record will be stored in the table "users_cars" joining the "user_id" with the selected "car_id". Initially arbitrary vales will be added for (car_latitude, car_longitude, service_Info) with initial values for (lock_status, engine_status). A response with OK will be returned to the user application to send him/her to the Home screen. The user will be able to log in, using his/her username and password, next time he/she uses the application to be authenticated by the backend.

The Home screen will be populated with information such as (car model, make and year) and a user can now update his/her information in the profile section or the service_Information.

The user will be able to send a request to fetch the last updated location of the vehicle(car_latitude, car_longitude) and perform other actions such as (Lock/Unlock car, Engine start/stop and Horn on/off). Figure 4.2 shows the data flow diagram.

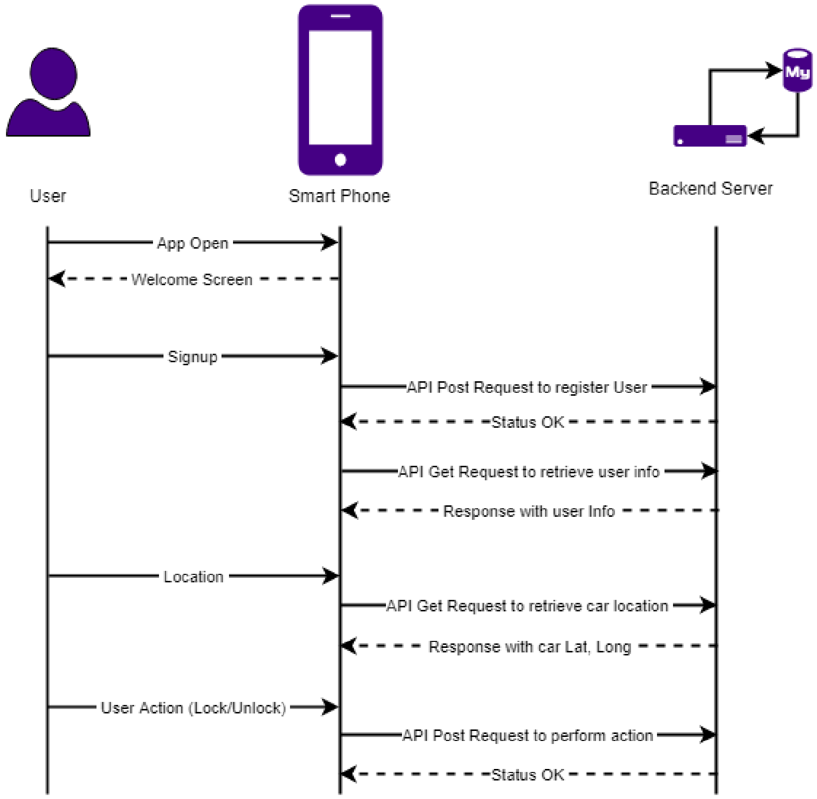

My
User
Smart Phone
Backend Server
App Open
Welcome Screen
Signup
API Post Request to register User
Status OK
API Get Request to retrieve user info
Response with user Info
Location
API Get Request to retrieve car location
Response with car Lat, Long
User Action (Lock/Unlock)
API Post Request to perform action
Status OK

Figure 4.2: Data Flow Diagram

Resources that are required for this simulation include:

- Apple iPhone 7
- Xcode program to develop the app in Apples Objective-C language
- Ubuntu Server, to represent the vehicle
- MySQL Database
- PHPMyAdmin GUI for MySQL server.
- Java Tomcat Application Server.

4.3 Mobile Application

Users start with a welcome screen to login or sign up. After the log-in, the user will be redirected to the Home screen where he/she can perform various actions, such as (Lock/Unlock car, locate the car or display the user information). After the user signs out, he/she will be returned back to the home screen. Figure 4.3 represents the application flow.

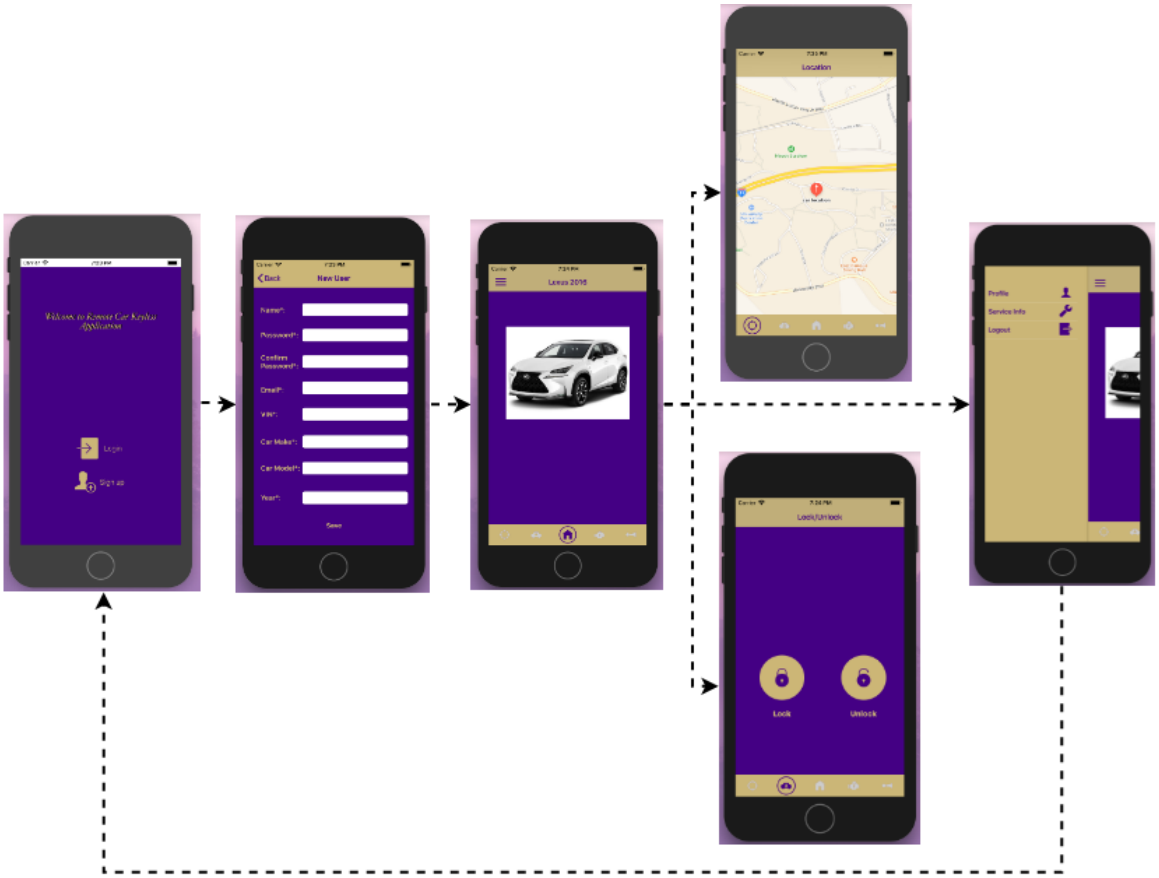


Figure 4.3: Application Flow Diagram

4.3.1 Welcome and Sign-Up Pages

First, the welcome page that has two main functionalities:

- Sign up where the user will be directed to a new page to sign up for a new account.
- Login where the user will be directed to a new page to login to his/her account using the password.

The Sign up page, gather all the required information from the user such as (name, password, email, car VIN number, Car Model, Car Make, Year) and send a POST request to the backend to save the information, then the page waits for the success response to redirect to the Home Page.

Figure 4.4 shows Welcome and Sign-Up pages in the application.

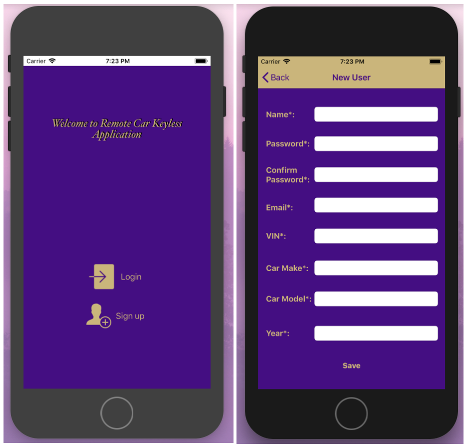


Figure 4.4: Welcome and Sign-Up pages

4.3.2 Home and User Information Side Pages

The user could slide from the left or push the button at the Home page and see a list of options

- Profile: the user is redirected to a page to view/edit the saved information.
- Service Info: the user is redirected to a page to fetch (GET request) the desired service information. The user can edit the service information, if necessary.
- Logout: The user is redirected to the welcome screen.

Figure 4.5 shows Home and User Information side pages in the application.

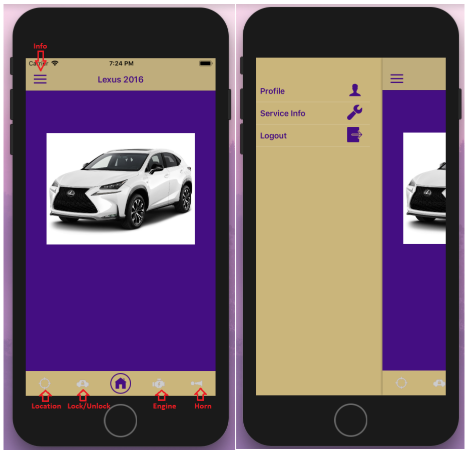


Figure 4.5: Home and User Information Side Pages

4.3.3 Location and Lock/Unlock Pages

The user can select the location icon, from the Home page bar to retrieve the car location, the location page is shown and a GET request is sent to the backend server to fetch the most updated Latitude and Longitude of the car and the map zooms to the vehicle's location. Also, the user can perform several functionalities from the app. The user can select the Lock/Unlock icon at the Home page to perform the required functionality. The Lock/Unlock page is displayed and a POST request is sent to the backend server to update the database with the selected functionality.

Despite the fact that more functionalities can be added such as open windows, open roof..etc, we decided to use a simple functionality such as the Lock/Unlock to perform several attacks as shown later in Chapter 5.

Figure 4.6 shows Location and Lock/Unlock pages in the application.

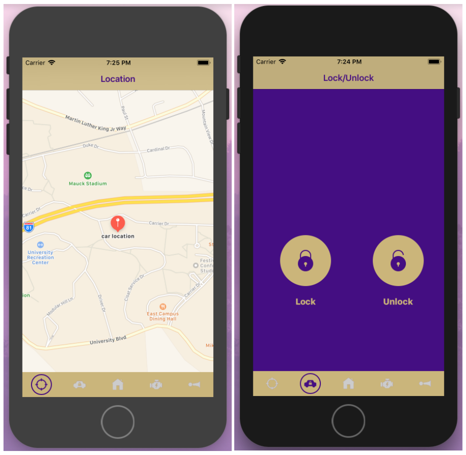
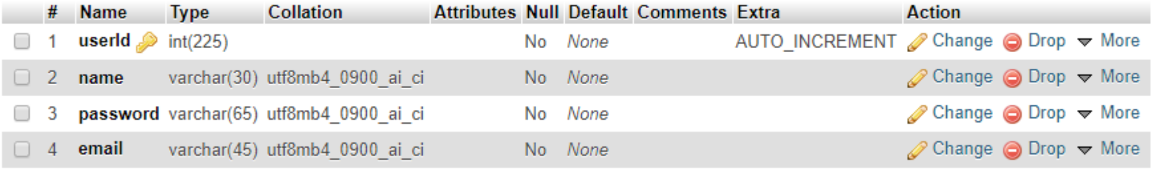


Figure 4.6: Location and Lock/Unlock pages

4.4 Database

The database is hosted on the host server which represents the vehicle. In a fully developed application, the database will be hosted on a Cloud-Storage such as Amazon Web Servers. The database is using MySQL with PHPMyAdmin. The database information is reflected on the mobile application, so the users can view/update this information. The following figures contain screenshots of the database configuration and how it was structured along with an Entity Relationship (ER) diagram.

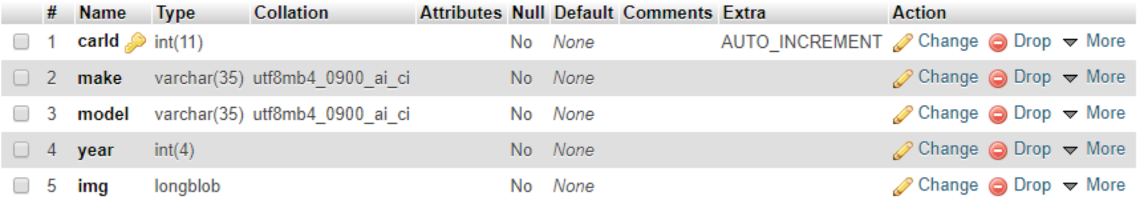
The user_id is the primary key of the table where it is an auto-increment value that is added the moment a new user registers. The name and email are the user's personal name and email address. Finally, the password is hashed in the Users Table to keep the privacy of the users. Figure 4.7 shows the users table.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**userId** 🔑	int(225)			No	*None*		AUTO_INCREMENT	Change ⊖ Drop ▼ More
□	2	**name**	varchar(30)	utf8mb4_0900_ai_ci		No	*None*			Change ⊖ Drop ▼ More
□	3	**password**	varchar(65)	utf8mb4_0900_ai_ci		No	*None*			Change ⊖ Drop ▼ More
□	4	**email**	varchar(45)	utf8mb4_0900_ai_ci		No	*None*			Change ⊖ Drop ▼ More

Figure 4.7: Users Table

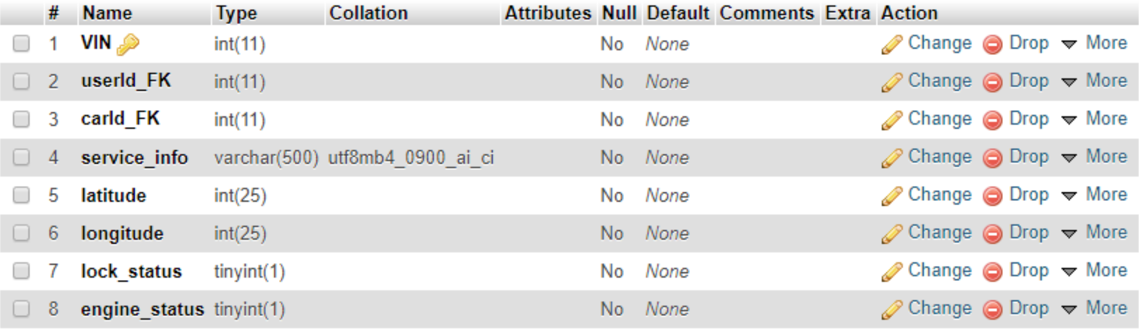
In the Cars Table, information about the vehicle's make, model, and year are stored. Also, an image of the vehicle model is stored in the database. Figure 4.8 shows the cars table.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**carId** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change ⊖ Drop ▼ More
□	2	**make**	varchar(35)	utf8mb4_0900_ai_ci		No	*None*			Change ⊖ Drop ▼ More
□	3	**model**	varchar(35)	utf8mb4_0900_ai_ci		No	*None*			Change ⊖ Drop ▼ More
□	4	**year**	int(4)			No	*None*			Change ⊖ Drop ▼ More
□	5	**img**	longblob			No	*None*			Change ⊖ Drop ▼ More

Figure 4.8: Cars Table

The fact that one user may have more than one vehicle and one vehicle can be owned by more than one user, made us design the database as a many-to-many relation M-to-M. In the Users-Cars Table, information such as VIN, service information, last known latitude and longitude, and lock_status are stored. More information can be added later such as engine_status and other information. Figure 4.9 shows the Users-Cars table.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**VIN** 🔑	int(11)			No	*None*			Change ⊖ Drop ▼ More
□	2	**userId_FK**	int(11)			No	*None*			Change ⊖ Drop ▼ More
□	3	**carId_FK**	int(11)			No	*None*			Change ⊖ Drop ▼ More
□	4	**service_info**	varchar(500)	utf8mb4_0900_ai_ci		No	*None*			Change ⊖ Drop ▼ More
□	5	**latitude**	int(25)			No	*None*			Change ⊖ Drop ▼ More
□	6	**longitude**	int(25)			No	*None*			Change ⊖ Drop ▼ More
□	7	**lock_status**	tinyint(1)			No	*None*			Change ⊖ Drop ▼ More
□	8	**engine_status**	tinyint(1)			No	*None*			Change ⊖ Drop ▼ More

Figure 4.9: Users-Cars table

Figure 4.10 shows the Entity Relationship diagram where two tables are connected using a M-to-M relationship.

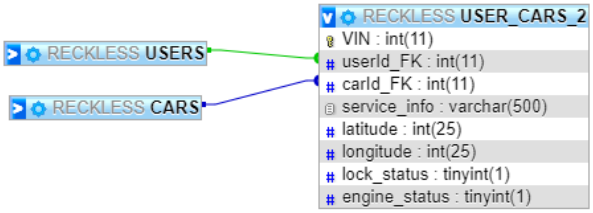


Figure 4.10: Entity Relationship Diagram

In this chapter, we were able to create a mobile application that simulates the real-application used to replace the remote keyless used these days to perform the needed functionality on a vehicle. This simulated app is used to perform a series of cyber-attacks as shown in Chapter 5.

Chapter 5

Security Attacks on RECKLESS-APPs

In Chapter 4, we developed a system that simulates the Remote Keyless Applications. An iOS mobile application was designed, developed and built to act similar to most of the applications in the app store. In addition, a Linux Server was built to act as the vehicle services and the vehicle computer. In this Chapter, we study the security attacks that can occur on vehicles and the mobile application, such as the Man in the Middle attack, Traffic Analysis attack, Social Engineering attack, Denial of Service attack, and etc. Then, we showed the types of attacks performed on our system. Finally, we analyze some of these cyber-attack vectors and provides mitigation techniques, cyber hygiene best practices, and suggest ways to balance security versus convenience.

5.1 Cyber Attacks on Vehicles and Applications

A number of cyber-attacks could be performed on both sides of the system: the vehicle and/or the Remote Keyless application. In this section, we discussed briefly classifications of cyber-attacks [37, 38]. We explain couple of these attacks and how it can be performed. We start with the *Monitoring Attacks*, which is the most common type of attacks related to our system. In these types of attacks, the attacker can track and monitor the system by monitoring the network communication, occurring between the vehicle and the application.

5.1.1 Man-in-the-Middle(MITM) Attack

Man-in-the-middle attack is a real time unauthorized interception of the private communication between the victim and his/her vehicle. The man-in-the-middle attack tries to gather as much possible information as possible from the connection, without the victim's knowledge [39, 40, 41]. The victim's information is then used to do another attack or to access the vehicle. Figure 5.1 shows the Man-in-the-Middle(MITM) attack.

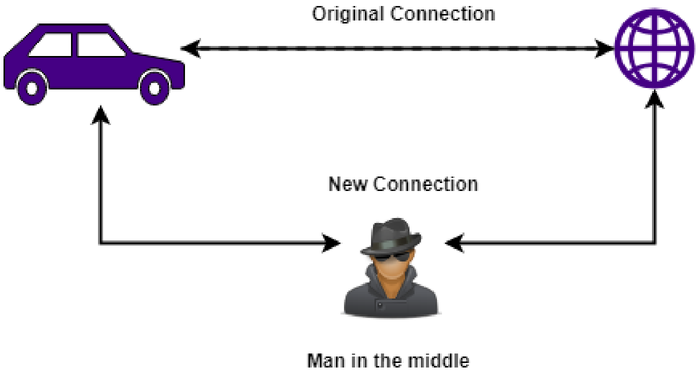
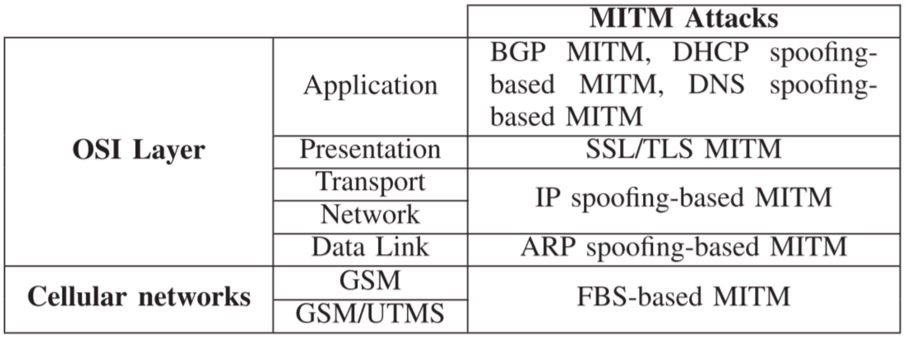


Figure 5.1: Man in the middle attack

Another example of an attack following the Man-in-the-Middle attack is to follow up with Traffic Analysis Attack. The attackers collect the information from the vehicle about network nodes identities or characteristics to determine and identify various kind of traffic information packets. Figure 5.2 shows MITM attack on different layers of the OSI Model and types of cellular networks [42].



		MITM Attacks
OSI Layer	Application	BGP MITM, DHCP spoofing-based MITM, DNS spoofing-based MITM
	Presentation	SSL/TLS MITM
	Transport	IP spoofing-based MITM
	Network	
	Data Link	ARP spoofing-based MITM
Cellular networks	GSM	FBS-based MITM
	GSM/UTMS	

Figure 5.2: MITM Attack on Different Layers of OSI Model and Types of Cellular Networks[42]

5.1.2 Social Engineering Attack

Social Engineering attack is the psychological manipulation of people into performing actions or making decisions that could affect the security or the safety of the victim's vehicle [43]. Attacker would send deceiving or even inappropriate messages to make the victim do things that would compromise the system's security or make the victim angry to behave irrationally [44]. There are several examples of attacks that use Social Engineering [45, 46].

5.1.3 Application-oriented Attack

In this type of attack, the attacker targets the vehicles safety applications which are deployed in a vehicle. They want to get information on events such as (car accident, traffic jam, post-crash, obstacle, emergency braking and so on)[47]. A customized version of this attack is the *Broadcast Tampering Attack*, where the attackers change the broadcasted safety messages to false safety messages. This tampering can causes more accidents and create huge traffic jams. A new type of application attacks is *Illusion Attack*, in which the attackers deceive the vehicle sensors and resulting in providing wrong traffic warning messages [47].

5.1.4 Denial of Service Attack

Denial-of-Service (DoS) attack target the availability of vehicle's service system, by flooding the system with unnecessary traffic packets to reduce the system availability diminish operation accuracy [37]. Several researchers predict that the next Denial of Service attacks on vehicles are imminent [48, 49]. Figure 5.3 shows the Denial of Service attack concept.

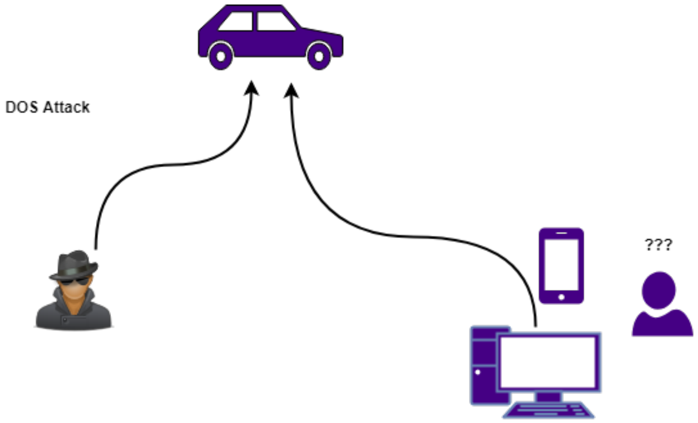


Figure 5.3: DOS Attack

5.1.5 Other Attacks

There are several other attacks that can occur in the vehicle world. For example, *Sybil Attack*, the attacker forges the identity of one or more vehicles. They deceive the other vehicles in the network to cast any type of attack on the system. In *Malware/Spamming Attacks*, attacker infects the vehicle with a worm or virus to send the critical information from the vehicle and to other servers. These attacks may be done by a fraudulent insider or by the outsiders[37]. A *Black Hole Attack*, makes a vehicle to decline to be a part in the network, when all network traffics is diverted to this particular vehicle, which causes the information to be dropped or lost[43].

5.2 Attacking RECKLESS-Apps

In the previous section 5.1, we discussed various classes and types of possible cyber-attacks on vehicles and applications. In this section, we discuss how cyber-attacks are performed on our system. In order to perform any kind of attack, we looked first at how the real keyless applications work and decided our cyber-attacks. Figure 5.4 shows the connection between the mobile application and the vehicle.

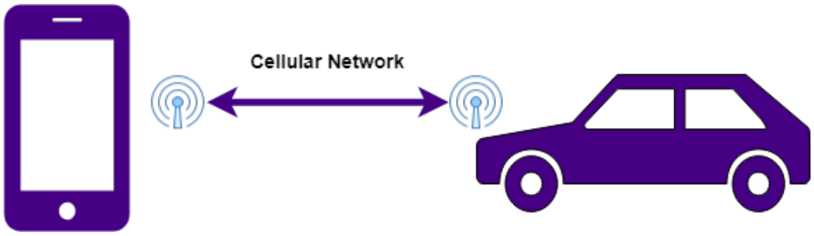
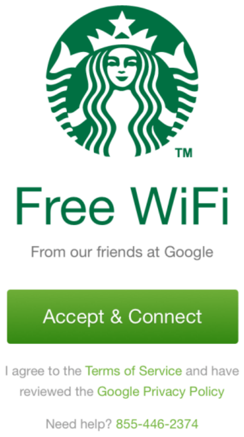


Figure 5.4: Connection between the mobile application and the vehicle

A Remote Keyless Mobile application typically connects to the vehicle over cellular network, at any place and any time to perform different functionalities that the application provides. However, it is also possible for the user to use WiFi to perform these functions. We apply the Evil Twin Attack, tricking our victim to use a malicious WiFi network that we created. In the following subsection we show the details Evil Twin attack.

5.2.1 Applying Evil Twin Attack

Most public places, coffee shops, fast food restaurants, and airports provide free WiFi network access to their customers, offering free Internet access. However, most of these types of networks are insecure, requiring no authentication and encryption. When the user accesses these WiFi networks, he/she must agree to their terms and conditions, where the Internet Service Provider(ISP) accepts no responsibility for the security or privacy of the customer information[50]. The Evil Twin Attack(ETA) allows the victim to access a fake WiFi network with a stronger signal. An ETA can be established by an attacker to mock the role of a legitimate WiFi access point and gain information/control without the victim's knowledge. An open WiFi network can only be recognized by its MAC address and its Service Set Identifier(SSID). This impersonation is relatively simple, since the attacker's fake access point (AP) may provide a more powerful signal to the victim, encouraging the victim to connect to the attacker's network instead of the legitimate network [51]. Figure 5.5 shows the real Starbucks WiFi and how it can be easily mocked.



Free WiFi

From our friends at Google

Accept & Connect

I agree to the Terms of Service and have reviewed the Google Privacy Policy

Need help? 855-446-2374

Figure 5.5: Starbucks Free WiFi

A victim, driver, decided to sit in a public area. The strong WiFi signal may encourage the victim to connect to the fake WiFi. The driver may use his/her application to warm the car, especially in winter time. Figure 5.6 shows the overview of this attack that we performed.

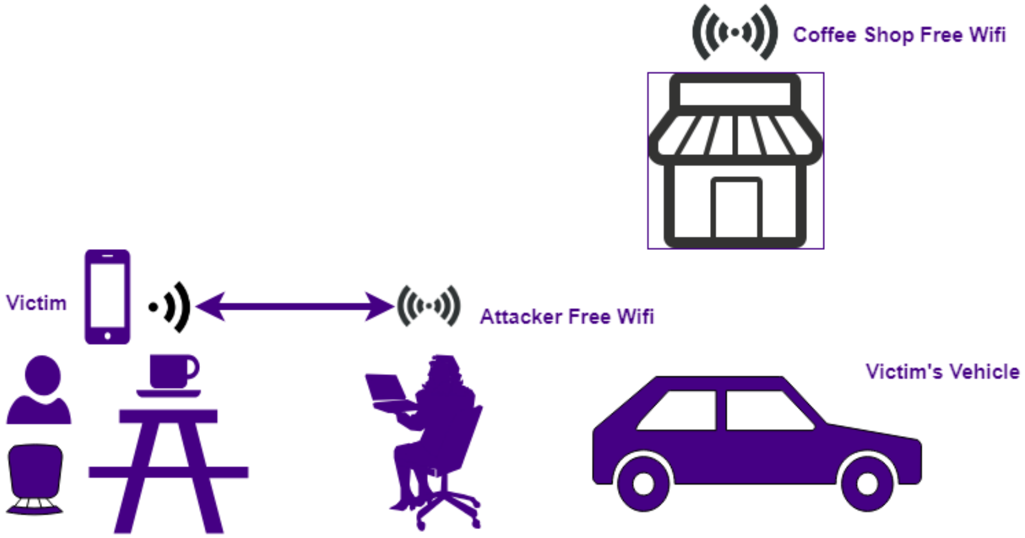


Figure 5.6: The victim is using the attacker's WiFi

After the victim connects to the attacker's AP, the attacker would spy on the victims wireless data and collect data that can be used to perform a number of attacks. Below, we listed different kinds of attacks that could be performed, after a successful Evil Twin Attack.

- **A Mobile Data Consumption Attack:** The first thing an attacker wants is to make sure that the victim doesn't switch back to the cellular network to use the Remote Keyless Application. The victim must keep on using the malicious WiFi, so the attacker can proceed with other attacks. According to Wasil [51], attacker can force the victims smartphone to consume data through the cellular network by switching the victim from the WiFi network to the cellular network while downloading a large file. This will make sure that the victim will subsequently depend on our malicious WiFi.

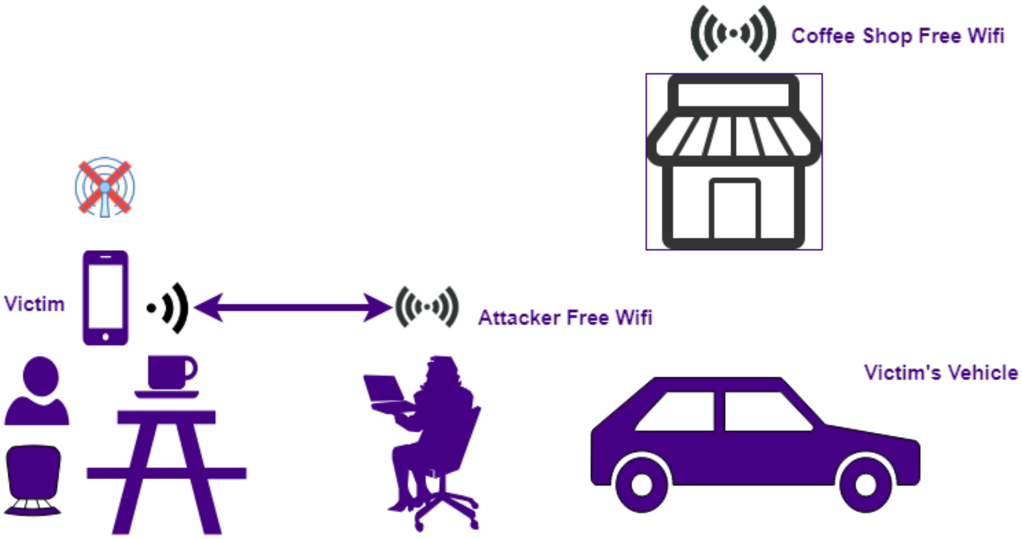


Figure 5.7: Mobile Data Consumption Attack

- **Man in the Middle Attack:** Man in the middle attack is the most relevant attack in our case, and it is the main target of an attacker after the success of the Evil Twin Attack. The attacker can use, for example, Address Resolution Protocol (ARP) spoofing/poisoning to complete a Man in the Middle attack. The ARP protocol is used to find MAC address of a host associated with an IP address on a subnet. When the hacker uses this technique, it can intercept the data in transit, compromising both the integrity and confidentiality of the system.

- **Denial of Service Attack:** Once the man in the middle attack was performed, and necessary information was gathered. The attacker will perform a Denial of Service attack on the victim's phone, by flooding it with more traffic than it can handle. The smartphone would remain connected to the fake WiFi without any useful data service. This may force the victim to leave the place [52].

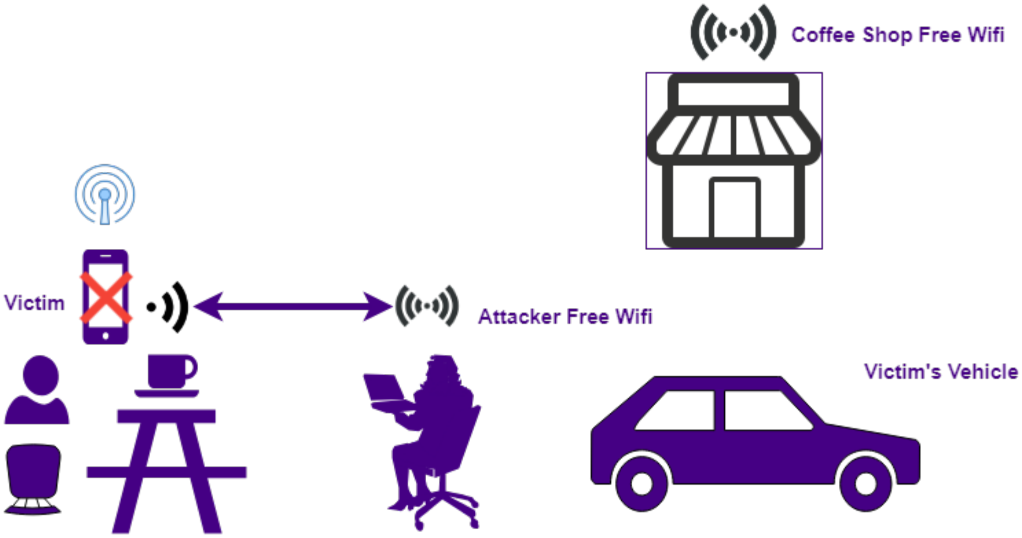


Figure 5.8: Denial of Service Attack

- **GPS spoofing Attack:** Once the man in the middle attack was performed, and necessary information was gathered. The attacker can perform attacks on the vehicle that was compromised. In a V2V network, vehicles depend on the integrity of data they receive from other vehicles and road side units to make decisions regarding safety messages and alerts. The attacker can create a GPS-simulator to generate false readings to deceive other vehicles in the network[53].

5.2.2 Applying Man in the Middle Attack

We performed the Man in the Middle Attack including ARP poisoning. We used two programs to perform our attack, the first program is called ”Cain & Abel” which runs on Windows Operating System. It allows sniffing the network traffic and cracking encrypted passwords.

Once you start the sniffer, you get a list of IP addresses, related to the clients on the network. Using the Victim IP and the Gateway IP (Malicious Router), attacker can start APR by click on the radioactive button in the toolbar.

The other tool we used is Wireshark. We started by selecting the network interface which in our case is the malicious WiFi (under name: Starbucks-Free), Wireshark inspects packets coming from the victim’s IP to make sure that the ARP poisoning is successful. Attacker can filter packets and identify the Remote Keyless Application as the host and get the user credentials.

5.2.3 Summary of Attacks

1. The victim joins the attacker's fraudulent WiFi network (Evil Twin Attack).
2. The attacker captures the victim's credentials when using the application (Man in the Middle Attack).
3. The attacker steals the vehicle using the victim's credentials.

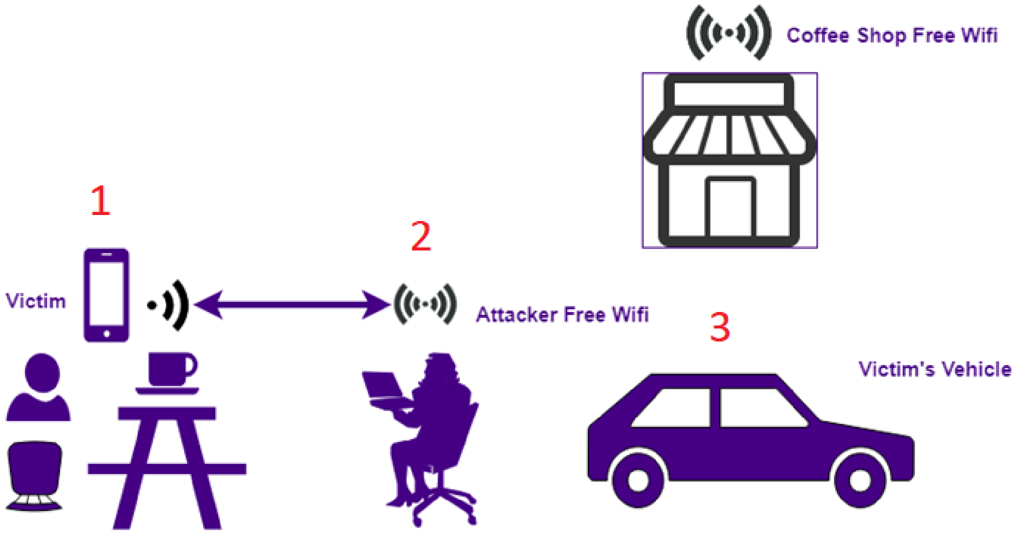


Figure 5.9: Summary of Attacks

5.3 Mitigation solutions and recommendations to protect RECKLESS-Apps from Cyber attacks

The whole idea of performing these cyber-attacks is to provide mitigation techniques, cyber hygiene best practices, and suggest ways to balance security versus convenience. In this section, we are considering the mitigation solutions and recommendations against these two attacks to our RECKLESS-App.

5.3.1 Evil Twin Attack Mitigation

In order to protect the system from Evil Twin attack, we present two methods of defense [54].

- **Context-leashing** is a detection strategy based on recording the nearby access points when first associating with an access point. Using this context, the victim can determine if an adversary did already setup an evil twin access point at a different location. This technique can be deployed without any modification to the access points, using only the contextual information, which is the list of all access points that are visible to the client at a familiar location. Let's say that a client associates to an access point with an SSID Fav-Coffee. While associated, the client can see messages from several other access points, indicating that there are other wireless networks nearby and those nearby SSIDs define the context for Fav-Coffee SSID. When the client observes the same wireless network with SSID Fav-Coffee at another location, such as railway station or airport, the client should be suspicious of associating to this access point as it might be an evil twin. The problem with this approach is that it does not provide any form of authentication or any confidentiality mechanism to prevent injection and eavesdropping which is the case in the following method EAP-SWAT.

- **EAP-SWAT** is a Secure Shell SSH-style authentication that performs one-way access point authentication, which fits into the extensible authentication protocol (EAP) framework. EAP is used to authenticate simple dialup and LAN connections. Its major scope is wireless network communication such as access points used to authenticate client-wireless/LAN network systems using a simple request and response mechanism [55]. EAP-SWAT provides one-way authentication mechanism that does not require any key exchange; However, since this authentication approach is based on the principle of trust-on-first-use, it could be vulnerable during the first authentication.

We recommend any user not to use any unknown unsecured WiFi because as we saw in the previous sections, it allows the data exchange without any form of encryption or security protection.

5.3.2 Man in the Middle Mitigation

To deter Man in the Middle attacks, data encryption must be used to ensure that intercepted traffic is difficult to decipher by the attacker.

5.4 Summary of Attacks and Mitigation

In this section, we summarizes the attacks and mitigation solutions and recommendations. In table 5.1,we show the recommendations for mitigation and protection on the two attacks we performed.

Table 5.1: Summary of attacks and mitigation

Attack	Mitigation
Evil Twin	1. context-leashing 2. EAP-SWAT 3. No connection to unsecured network
Man in the Middle	Encrypt the traffic

Chapter 6

Conclusion and Future Work

In this chapter, we conclude our work and discuss future research directions that could be built on our current work.

6.1 Conclusion

In chapter 2 we provided related research, background information and research work to support different components of our thesis as well as highlighting the overall history of keys used for vehicles. This study helped us to understand how the current Remote keyless system works and how it was evolved. In Chapter 3, we showed the analysis and design of the Remote Car keyless Applications on smartphones. We discussed several existing applications in the market and made a comparison between the features. We described the architecture diagram of our simulation. In Chapter 4, we showed the sequence diagram of the iOS application we built for the simulation and its communication with the backend server. In Chapter 5, we showed types of attacks performed on our system (Evil Twin Attack and Man in the Middle attack). Finally, we analyzed some of these cyber-attack vectors and provided mitigation techniques.

In Chapter 5, we concluded how to protect the user from cyber attacks that may happen while using the Remote Keyless Application. The applications development engineers should design these applications to communicate through secure channels such as SSL, TLS, and HTTPS. These secure techniques can mitigate the Man in the Middle attacks. There are many applications in the app store that are not using secure communication and are vulnerable to cyber attacks. The user must avoid connecting to any unsecured WiFi connection, to avoid the Evil Twin Attack. There should be a warning in the welcome screen advising the user to avoid using this application on an unsecured WiFi connection and advise the user to use the cellular network instead.

6.2 Future Work

In this section, we describe future research directions to extend our work with more studies about the cyber attacks that could affect the Remote Car Keyless applications. We summarize future work in the following points:

- Instead of deceiving the victims with the malicious WiFi connection (Evil Twin Attack), capture the cellular network directly using RTL-SDR. Using this tool attacker can sniff the unencrypted data.

- Capture the bluetooth signals from a remote cell phone and transmit it to the vehicle via gadgets and study the possibility of deceiving the vehicle.

- Use HTTPs in the communication between the application and the backend server and study the decryption of traffic using the Wireshark.

- Study different cyber attacks on vehicles after compromising the communication between the application and the vehicle. Attacks such as (Timing Attack, Tunneling attack, Masquerade attack and Replay attack) and show how these attacks could affect the ITS.

Bibliography

[1] "Bmw says car keys may be replaced by mobile phone apps," Sep 2017. [Online]. Available: https://www.reuters.com/article/us-autoshow-frankfurt-keys/bmw-says-car-keys-may-be-replaced-by-mobile-phone-apps-idUSKCN1BQ1ES

[2] M. Molloy, "Mercedes car stolen without using a key in seconds in 'relay theft'," Nov 2017. [Online]. Available: https://www.telegraph.co.uk/news/2017/11/27/mercedes-car-stolen-without-using-key-seconds-relay-theft/

[3] U. D. of Transportation, "Intelligent transportation systems," Online, 2018. [Online]. Available: https://www.its.dot.gov

[4] C. Weiß, "V2x communication in europe–from research projects towards standardization and field testing of vehicle communication technology," *Computer Networks*, vol. 55, no. 14, pp. 3103–3119, 2011.

[5] U. D. of Transportation, "Standard specification for tele- communications and information exchange between roadside and vehicle systems - 5 ghz band dedicated short range communications (dsrc) medium access control (mac) and physical layer (phy) specifications," in *United States Patent No. 772364*, Aug 2003, pp. ASTM E2213–03.

[6] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, nov. 2007, pp. 46 –51.

[7] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," *Robotics and Autonomous Systems*, vol. 62, no. 1, pp. 16 – 24, 2014, new Boundaries of Robotics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889012001236

[8] J. Barrachina, J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J. C. Cano, C. T. Calafate, and P. Manzoni, "V2x-d: A vehicular density estimation system that combines v2v and v2i communications," in *2013 IFIP Wireless Days (WD)*, Nov 2013, pp. 1–6.

[9] X. Wu, R. Miucic, S. Yang, S. Al-Stouhi, J. Misener, S. Bai, and W.-h. Chan, "Cars talk to phones: A dsrc based vehicle-pedestrian safety system," in *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*. IEEE, 2014, pp. 1–7.

[10] P. Rahimian, E. E. ONeal, S. Zhou, J. P. Yon, L. Franzen, J. M. Plumert, and J. K. Kearney, "Vehicle-to-pedestrian (v2p) communications technology: Do cell phone warnings improve road-crossing safety for texting pedestrians?" The University of Iowa, Tech. Rep., 2018.

[11] R. Bishop, "A survey of intelligent vehicle applications worldwide," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, 2000, pp. 25–30.

[12] C. X. Bo Yu, "Vehicular ad-hoc networks: An information-centric perspective," *ZTE Communications*, vol. No.3, 2010.

[13] M. Lake, "How it works; remote keyless entry: Staying a step ahead of car thieves," Jun 2001. [Online]. Available: https://www.nytimes.com/2001/06/07/technology/how-it-works-remote-keyless-entry-staying-a-step-ahead-of-car-thieves.html

[14] R. Kayne and N. Foster, "What is keyless entry?" Apr 2018. [Online]. Available: http://www.wisegeek.org/what-is-keyless-entry.htm

[15] "Here's how a 17 gadget breaks your car's keyless system." [Online]. Available: http://www.financetwitter.com

[16] S. Pocket, "Key fob guards / key fob protector for rfid key fobs." [Online]. Available: https://silent-pocket.com/products/key-fob-guards

[17] urkkik, "Tesla car remote control is difficult to use. look why." Sep 2016. [Online]. Available: https://www.youtube.com/watch?v=KyTwyyvwQf8

[18] A. Misra, A. Gooze, K. Watkins, M. Asad, and C. Le Dantec, "Crowdsourcing and its application to transportation data collection and management," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2414, pp. 1–8, 2014.

[19] S. Lee, G. Tewolde, and J. Kwon, "Design and implementation of vehicle tracking system using gps/gsm/gprs technology and smartphone application," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 353–358.

[20] R. Araújo, Â. Igreja, R. de Castro, and R. E. Araujo, “Driving coach: A smartphone application to evaluate driving efficient patterns,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE.* IEEE, 2012, pp. 1005–1010.

[21] M. Garcia, P. Rose, R. Sung, and S. El-Tawab, “Secure smart parking at james madison university via the cloud environment (space),” in *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, April 2016, pp. 271–276.

[22] Y. R. Rao, “Automatic smart parking system using internet of things (iot),” *Int J Eng Technol Sci Res*, vol. 4, no. 5, 2017.

[23] C. Miller and C. Valasek, “A survey of remote automotive attack surfaces,” *black hat USA*, vol. 2014, 2014.

[24] D. A. Johnson and M. M. Trivedi, “Driving style recognition using a smartphone as a sensor platform,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.* IEEE, 2011, pp. 1609–1615.

[25] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, “Wreckwatch: Automatic traffic accident detection and notification with smartphones,” *Mobile Networks and Applications*, vol. 16, no. 3, p. 285, Mar 2011. [Online]. Available: https://doi.org/10.1007/s11036-011-0304-8

[26] G. Castignani, T. Derrmann, R. Frank, and T. Engel, “Driver behavior profiling using smartphones: A low-cost platform for driver monitoring,” *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 91–102, 2015.

[27] Tesla, “Tesla application,” Online, 2018. [Online]. Available: https://itunes.apple.com/us/app/tesla/id582007913?mt=8

[28] BMW, “Bmw connected application,” Online, 2018. [Online]. Available: https://itunes.apple.com/us/app/bmw-connected/id1087277146?mt=8

[29] Jeep, “Jeep uconnect application,” Online, 2018. [Online]. Available: https://itunes.apple.com/us/app/uconnect/id1229236724?mt=8

[30] “Applications.” [Online]. Available: http://www.remowireless.com/en/applications/

[31] H. Osborne, “Thieves target luxury range rovers with keyless locking systems,” Oct 2014. [Online]. Available: https://www.theguardian.com/money/2014/oct/27/thieves-range-rover-keyless-locking

[32] J. Steinberg, "Vulnerability in car keyless entry systems allows anyone to open and steal your vehicle," May 2015. [Online]. Available: https://www.forbes.com/sites/josephsteinberg/2015/05/12/vulnerability-in-car-keyless-entry-systems-allows-anyone-to-open-and-steal-your-car/#4d83e63c2442

[33] J. Wright, M. E. Dawson Jr, and M. Omar, "Cyber security and mobile threats: The need for antivirus applications for smart phones," *Journal of Information Systems Technology and Planning*, vol. 5, no. 14, pp. 40–60, 2012.

[34] "Mapkit." [Online]. Available: https://developer.apple.com/documentation/mapkit

[35] "Object-oriented programming with objective-c," Nov 2010. [Online]. Available: https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/OOP_ObjC/Articles/ooWhy.html

[36] S. Dimatteo, P. Hui, B. Han, and V. O. Li, "Cellular traffic offloading through wifi networks," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on.* IEEE, 2011, pp. 192–201.

[37] A. Quyoom, R. Ali, D. N. Gouttam, and H. Sharma, "A novel mechanism of detection of denial of service attack (dos) in vanet using malicious and irrelevant packet detection algorithm (mipda)," in *International Conference on Computing, Communication Automation*, May 2015, pp. 414–419.

[38] N. Hussain, A. Singh, and P. K. Shukla, "In depth analysis of attacks & countermeasures in vehicular ad hoc network," *International Journal of Software Engineering and Its Applications*, vol. 10, no. 12, pp. 329–368, 2016.

[39] I. A. Sumra, H. B. Hasbullah, and J.-l. B. AbManan, "Attacks on security goals (confidentiality, integrity, availability) in vanet: a survey," in *Vehicular Ad-Hoc Networks for Smart Cities.* Springer, 2015, pp. 51–61.

[40] K. Evers, R. Oram, S. El-Tawab, M. H. Heydari, and B. B. Park, "Security measurement on a cloud-based cyber-physical system used for intelligent transportation," in *Vehicular Electronics and Safety (ICVES), 2017 IEEE International Conference on.* IEEE, 2017, pp. 97–102.

[41] A. Wasicek, P. Derler, and E. A. Lee, "Aspect-oriented modeling of attacks in automotive cyber-physical systems," in *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE.* IEEE, 2014, pp. 1–6.

[42] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.

[43] I. A. Sumra, H. B. Hasbullah, I. Ahmad, D. M. Alghazzawi *et al.*, "Classification of attacks in vehicular ad hoc network (vanet)," *International Information Institute (Tokyo). Information*, vol. 16, no. 5, p. 2995, 2013.

[44] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2010, pp. 447–462.

[45] F. Mouton, M. M. Malan, L. Leenen, and H. S. Venter, "Social engineering attack framework," in *Information Security for South Africa (ISSA), 2014*. IEEE, 2014, pp. 1–9.

[46] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and applications*, vol. 22, pp. 113–122, 2015.

[47] N. W. Lo and H. C. Tsai, "Illusion attack on vanet applications - a message plausibility problem," in *2007 IEEE Globecom Workshops*, Nov 2007, pp. 1–8.

[48] T. S. Perry, "Why the next denial-of-service attack could be against your car," Oct 2016. [Online]. Available: https://spectrum.ieee.org/view-from-the-valley/transportation/safety/why-the-next-denial-of-service-attack-could-be-against-your-car

[49] A. Quyoom, R. Ali, D. N. Gouttam, and H. Sharma, "A novel mechanism of detection of denial of service attack (dos) in vanet using malicious and irrelevant packet detection algorithm (mipda)," in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. IEEE, 2015, pp. 414–419.

[50] O. Nakhila, E. Dondyk, M. F. Amjad, and C. Zou, "User-side wi-fi evil twin attack detection using ssl/tcp protocols," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 239–244.

[51] D. Wasil, O. Nakhila, S. S. Bacanli, C. Zou, and D. Turgut, "Exposing vulnerabilities in mobile networks: A mobile data consumption attack," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 550–554.

[52] E. Dondyk and C. C. Zou, "Denial of convenience attack to smartphones using a fake wi-fi access point," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Jan 2013, pp. 164–170.

[53] B. K. Bhargava, A. M. Johnson, G. I. Munyengabe, and P. Angin, “A systematic approach for attack analysis and mitigation in v2v networks,” *JoWUA*, vol. 7, pp. 79–96, 2016.

[54] K. Bauer, H. Gonzales, and D. McCoy, “Mitigating evil twin attacks in 802.11,” in *Performance, computing and communications conference, 2008. IPCCC 2008. IEEE International.* IEEE, 2008, pp. 513–516.

[55] “What is the extensible authentication protocol (eap)? - definition from techopedia.” [Online]. Available: https://www.techopedia.com/definition/5058/extensible-authentication-protocol-eap