# Crypto-analysis Resistant Digital Key FOB

Jeremy Hein
Adriana Matos
Courtney Eaton

# CONCEPT OF OPERATIONS

REVISION – 1a
September 19, 2023

# Concept of Operations
for
# Crypto-analysis Resistant Digital Key FOB

Team <2>

Approved by:

__________________________________
Project Leader Date

__________________________________
Prof. Kalafatis Date

__________________________________
T/A Date

## Change Record

| Rev. | Date | Originator | Approvals | Description |
|---|---|---|---|---|
| **-** | 9/4/2023 | Jeremy Hein | | Draft Release |
| **1** | 9/14/2023 | Courtney Eaton | | Submitted Revision |
| **1a** | 9/19/2023 | Courtney Eaton | | Updated Draft for Midterm Report |

# Table of Contents

# List of Figures

# 1. Executive Summary

Cybersecurity is becoming a larger issue as nearly all modern, everyday devices contain internet communication capabilities. With increasing dependence on dynamically re-designable software (i.e. software-defined vehicles) and increasingly connected vehicles with autonomous driving capabilities built-in, cybersecurity threats are no longer bound to information but extend out to physical properties and the safety of human beings. Current de facto ways of protecting the SDI (i.e. firmware updates) are Public Key Infrastructure (PKI) for authentication and integrity checks and Key Management System (KMS) for confidentiality. Unfortunately, both of these security services are very expensive to deploy and maintain, as well as being the lowest-hanging fruit/attack surface for malicious hackers. In response to this, Sandia National Laboratories has developed a new way to securely communicate and update firmware with a vehicle by replacing PKI and KMS with dynamic key generation and Zero Trust Architecture. The purpose of our project is to show the capabilities of this technology through the creation of a digital key FOB application that will utilize this new technology in order for the user to securely lock, unlock, and start their car. The applications of this new technology are abundant and it provides the opportunity for a new level of security within the vehicles of the future.

# 2. Introduction

The purpose of this document is to provide a conceptual framework for building a secure digital key FOB capable of resisting all forms of crypto-analysis attacks. Through a digital key FOB application, this project implements the Zero Trust Application for Vehicle technology (ZAV) which is made to be resistant towards Crypto-analysis attacks against one's vehicle. This project will have the potential to provide a secure communication channel, resistant to cyber attacks (i.e. relay, man-in-the-middle, crypto-analysis attacks, etc.) between the digital key FOB and the vehicle. This ZAV technology can easily be modified to be integrated as a secure way to update firmware in a vehicle Over-the-Air (OTA).

## *2.1 Background*

Many new automobiles now have a keyless entry instead of manually unlocking a car by pressing a button on a key FOB or turning a key. This keyless entry works by having a proximity key FOB with a short-range radio transmitter/radio frequency identification (RFID) chip and antenna that sends out a distinct rotating coded signal to a receiver unit in the car. When a driver, for example, pushes a button on the door handle, the system will transmit a signal to the authorized key FOB to see if it is within range. If the receiver is within range, the key FOB will then respond with its own signal to open the door.[3] This proximity key FOB will only unlock the car if it is close enough on the outside of the car and a button on the door handle is pressed to send out the signal. This method, although relatively secure, is not impervious to relay attacks.

Relay attacks are a form of automotive attack that requires two people and a relay device. One person stands by the automobile with the first relay device and triggers the car to send a signal out to confirm the key FOB is within range. Another person will stand near wherever the key FOB is with the second device. The second device will then transmit the relayed signal to the first device. Once the vehicle receives the signal, it will respond in a normal manner as if the key FOB is near and allow entry into the vehicle. To turn on the automobile, a second relay is needed using essentially the same process. [2]

This relay attack sparked many concerns within the automotive industry, which led to the creation of the Zero Trust Application (ZTA). Sandia National Laboratories patented the Zero Trust Application for Vehicle technology, or ZAV, which is a solution to many car hacking problems. The ZAV technology is a new secure communication that can provide all security attributes namely: confidentiality, integrity, authenticity, nonrepudiation, and availability without the use or reliance on PKI and KMS.

Using a similar algorithm, the digital key FOB will show how the ZAV technology can create a secure connection to communicate between the device application and the car. Once proven to be secure through the digital key FOB, this technology can be adapted to update firmware and protect the vehicle from future attacks.

## 2.2 Overview

The project will consist of an Android application that will connect to a physical locking mechanism through Bluetooth using a secured communication channel. The Android application will have the capability to securely register the user to a specific car through the use of a unique signature which will be inputted with the user's finger or stylus. Once registered, the user will be able to lock/unlock and turn on/off their car through the application and the microprocessor will use the unique digital behavior of the key FOB (i.e. mobile device, smartphone, tablet) to ensure that the device giving the command to the car is a physically designated key FOB.

The locking mechanism will consist of a student-built microprocessor with added CAN bus capabilities to mimic an automobile's internal network system. Using the signal sent from the application and the encryption/decryption software, a secure communication line will be created between the application and the locking mechanism. The CAN bus network will ensure that the signal is coming from the owner's device and turn on the necessary Electronic Control Units. Theoretically, if the signal is not coming from the owner's device the locking mechanism will not unlock showing that the message is not replayable, or cryptographically decipherable.

## 2.3 Referenced Documents and Standards

**Referenced Documents:**

[1] Choi, Peter. *Zero Trust Application for Vehicle*, Sandia National Laboratories, 2022.

[2] "Keyless Car Theft: What Is a Relay Attack, How Can You Prevent It, and Will Your Car Insurance Cover It?" *Leasing.com*, leasing.com/guides/relay-car-theft-what-is-it-and-how-can-you-avoid-it/.

[3] Shah, Vivek. "Proximity Keys and Other Unlocking Innovations." *CarExpert*, 24 Sept. 2021, www.carexpert.com.au/car-news/proximity-keys-and-other-unlocking-innovations. Accessed 9 Sept. 2023.

**Standards:**

- IEEE 802.15.1-2002
- IEEE 1363-2000
- IEEE 1363.3-2013
- IEEE 1625-2004
- NIST Cryptographic Standards and Guidelines

# 3. Operating Concept

## *3.1 Scope*

The scope of this project is to create a proof of concept build that will showcase the capabilities of Zero Trust Authentication in Vehicle safety and security applications through a digital key FOB demonstration. This will be accomplished with an Android application, in-depth backend encryption for commands sent through Bluetooth, and a microprocessor to act as the car and interpret the signals. The microprocessor will communicate with a locking and an ignition representation to indicate the success of the digital key FOB.

## *3.2 Operational Description and Constraints*

The Crypto-analysis Resistant Digital Key FOB is designed for widespread application across the transportation industry. To use the digital key FOB, the user will open the app and register the device as a digital key FOB with the vehicle ECU. Registration can be achieved by drawing or signing their name on a digital keypad. The device will pair with the car through a combination of the Advanced Encryption Standard (AES) algorithm and the Diffie-Hellman Key Exchange protocol. The user will then be able to send open and start commands through the app to the car, which will be represented by a microcontroller for this project.

The Android app will be developed through Android Studio using a mix of Kotlin, Java, and C++ for the UI and back-end development. The app will use a Bluetooth signal to send data packets to the microcontroller. Once a signal is received, the microcontroller will then use the CAN BUS architecture to send commands to a representative car lock and engine ignition devices.

## *3.3 System Description*

The Crypto-analysis Resistant Digital Key FOB will be made up of an android application, an encryption system that handles all communication, and a microprocessor**.** These requirements can be separated into the following subsystems.

**Android Application Subsystem:**
The Android application subsystem will integrate with the communication & cryptography subsystem to send Bluetooth data packets to the microcontroller. The application will take in the digital signature of the user by using a hand gesture or stylus and provide that data to the communication subsystem. The app will also take in the user commands and relay them to the communication & cryptography subsystem which will facilitate the communication between the mobile device and ECU.

**Locking Mechanism Subsystem - Hardware:**
This hardware subsystem will include a power source, a microprocessor, an LED (as the lock), the CAN bus network, and a Bluetooth connection. The microprocessor will provide the necessary components to help the CAN bus network run as needed. An ESP32 chip will be used to apply the CAN bus architecture onto the microprocessor and provide the Bluetooth connection

**Figure 1:** LEDs

to the application. Two LEDs will be connected to the CAN bus network and function as Electronic Control Units; one green LED to show the system is on and working and one red LED to show whether the system is locked or unlocked.

**Communication & Cryptography Subsystem:**
The communication and cryptography subsystem will cover the registration and communication processes between the device application and the car. During the registration process, the Z-Hardware Profile (a unique combination of the human signature entered by the user and a pseudo-random number generated by the ZAV mobile application) will be sent to the car with the use of the AES-256 Algorithm where the symmetric key will be created using the Diffie-Hellman Key Exchange between the car and device. During the communication process, the device uses a non-repeating timestamp, the Z-Hardware Profile, and “lock/unlock” commands to generate the Secure Hash Algorithm (SHA-256) dynamic digital signature. The car will then make its own dynamic digital signature using clear text timestamp and a “lock/unlock” command sent by the digital key FOB and its own stored Z-Hardware Profile to verify that the device is the correct device before accepting the command.

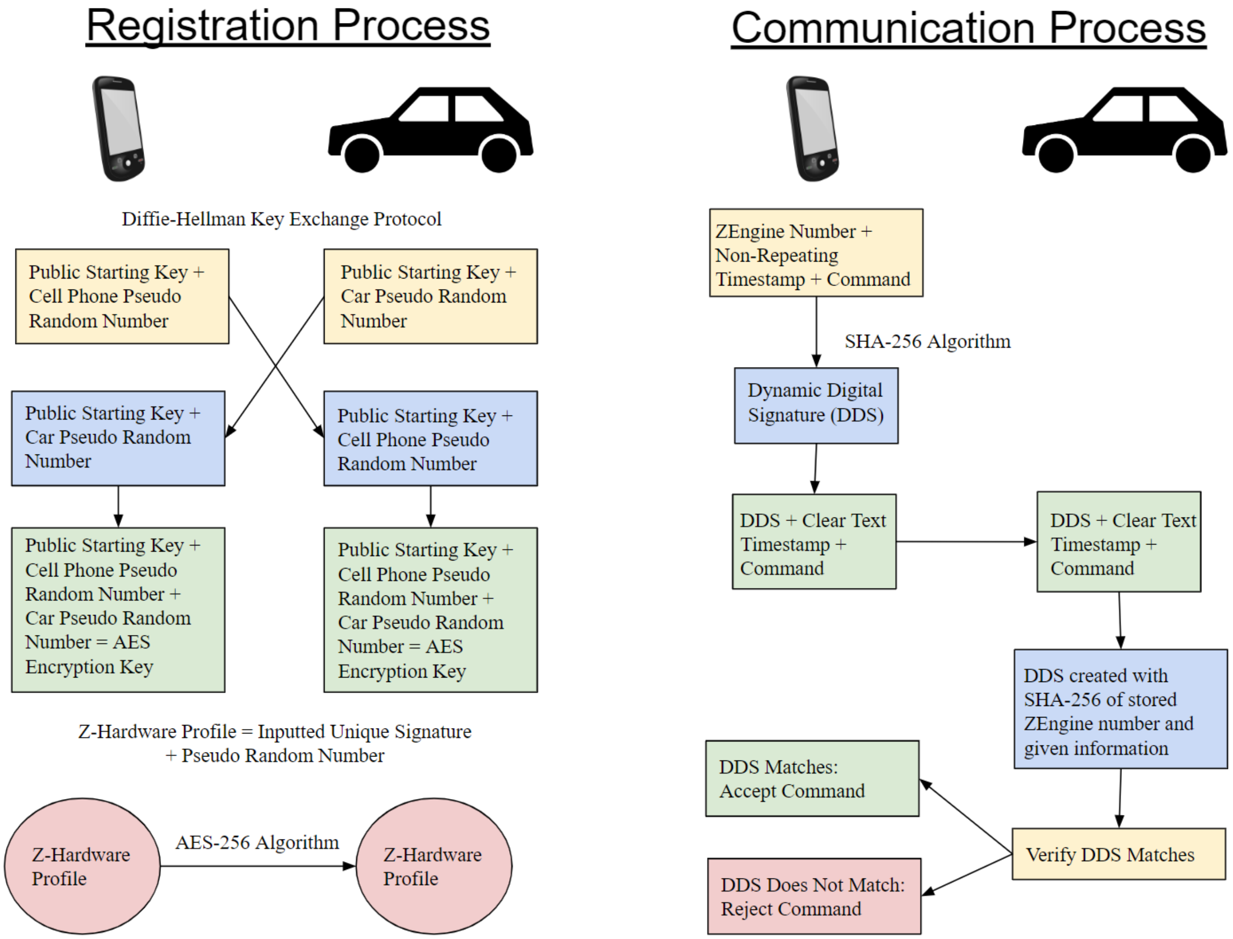
Registration Process
Diffie-Hellman Key Exchange Protocol
Public Starting Key + Cell Phone Pseudo Random Number
Public Starting Key + Car Pseudo Random Number
Public Starting Key + Car Pseudo Random Number
Public Starting Key + Cell Phone Pseudo Random Number
Public Starting Key + Cell Phone Pseudo Random Number + Car Pseudo Random Number = AES Encryption Key
Public Starting Key + Cell Phone Pseudo Random Number + Car Pseudo Random Number = AES Encryption Key
Z-Hardware Profile = Inputted Unique Signature + Pseudo Random Number
Z-Hardware Profile
AES-256 Algorithm
Z-Hardware Profile
Communication Process
ZEngine Number + Non-Repeating Timestamp + Command
SHA-256 Algorithm
Dynamic Digital Signature (DDS)
DDS + Clear Text Timestamp + Command
DDS + Clear Text Timestamp + Command
DDS created with SHA-256 of stored ZEngine number and given information
Verify DDS Matches
DDS Matches: Accept Command
DDS Does Not Match: Reject Command

**Figure 2:** Flow Diagram of Registration and Communication Processes

## *3.4 Modes of Operations*

**Registration:** The application will enter the registration mode when the user inputs their digital signature into the system, commencing the process for the application and the car microprocessor to exchange the Z-Hardware Profile, and replicating the Z-Hardware Profile for future communication between the the digital key FOB and ECU.

**Active:** The application will enter the active mode automatically after registration while it waits for the user to select either the lock/unlock or start/turn off car commands.

**Command**: The application will enter the command mode when a command is selected on the application. This will initiate the communication process between the device and the car where the device will create a dynamic digital signature and, along with the command, send it to the car where the car will verify the dynamic digital signature before accepting the command.

## *3.5 Users*

This digital key solution is intended for use by an everyday user who owns a motorized vehicle and who knows how to use basic digital applications. The Android application will be designed for ease of use and will require no training for operation. The end user will be able to interpret the button commands shown in the application and use them effectively.

## *3.6 Support*

Support will be given to users through a FAQ page in the app showing basic information on app navigation, possible troubleshooting tips, and contact information for technical support services. Since the user will not be able to access the code for the device, explanations for the inner workings of the application will not be provided.

# 4. Threat Modeling Scenarios & Countermeasures

Below are three possible scenarios for the ZAV technology. The first scenario is the main one that our project will be focusing on, specifically in being able to securely lock, unlock, start, and turn off a car. Once this project is completed, and it shows the capabilities of the ZAV technology, scenarios two and three highlight other scenarios in which this technology can be applied.

## *4.1 Anti-Theft Automobiles*

Automobile theft is nothing new, but as hackers are able to competently breach older methods of security, new methods are beginning to arise. The ZAV technology provides a solution to this auto theft problem due to its secure encryption/decryption software. Using ZAV and its secure, unique signal, thieves will no longer be able to hack into a car by relaying your key's (whether they be physical or digital) RFID signal.

## *4.2 Secure Cloud Protection*

Since the Zero Trust Application technology is a secure encryption/decryption software that can generate cryptographic keys on the fly, it is able to store sensitive data onto a cloud-based server and secure that data with the Zero Trust Application without the need for insecure KMS. Having the Zero Trust Application on a personal device will allow only that device to access the sensitive data and no one else. Even if the personal device is hacked, the hacker still will not be able to access the data saved on the cloud if it incorporates what you know into the mix of non-repeating input data for the cryptography and communication subsystem to generate the decryption key.

## *4.3 Security of Firmware Updates*

Most modern automobiles have increased connectivity and active sensing functionality integrated into the vehicle with about 200 million lines of code. Due to the scale and complexity of the coding, frequent software/firmware updates are needed but this leads to hackers being able to infiltrate firmware updates as they are sent over the air to the car. Using the ZAV technology, digital car components can be updated and "will be able to authenticate, encrypt/decrypt, and verify the integrity of 'the' message all in one signal data packet message"[1], thus only allowing firmware updates from the trusted source as well as guaranteed end-to-end encryption/confidentiality.

# 5. Analysis

## *5.1 Summary of Proposed Improvements*

- Utilization of a digital key FOB allows for the user to always have their key on them preventing misplacing or theft of their car key FOB
- Dynamically changing signals when sending commands prevents relay attacks
- One-time registration creates secure communication between the car and the application
- Application user does not have to manually verify their identity through the use of passwords or biometric information

## *5.2 Disadvantages and Limitations*

- Does not protect against hardware CAN bus injection attacks
- Not built to withstand any infiltration into the software.
- Recreating already existing software so will not completely match the technology of Sandia National Laboratories or fully match the capabilities of ZAV

## *5.3 Alternatives*

- PKI and AWS KMS
    - Popular infrastructure that is widely used
    - Less secure with many well-known strategies already in place to hack them
    - Requires the user to remember a password or input their biometrics
- Digital Key FOB provided by Car's Dealership
    - Already connected and compatible with the user's car
    - Can provide more features than just simple locking/unlocking and turning on/off the car
    - Does not provide the same level of security as the Crypto-analysis Resistant Digital Key FOB
- Hardware Key FOB
    - Comes with a key within it to allow for physical locking of the glove box and driver's side door
    - Easy to lose the key FOB or be stolen
    - Is susceptible to car theft through the use of a relay box to capture the signal and transmit it over to the car

## *5.4 Impact*

This project does not impact the environment or have any ethical concerns.

# Crypto-analysis Resistant Digital Key FOB

Adriana Matos
Courtney Eaton
Jeremy Hein

# FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 1
September 26, 2023

# Functional System Requirements

for

# Crypto-analysis Resistant Digital Key FOB

Prepared by:

Project Team 9/26/2023

Approved by:

Adriana Matos Date

John Lusher, P.E. Date

T/A Date

# Change Record

| Rev. | Date | Originator | Approvals | Description |
|---|---|---|---|---|
| **-** | 9/18/2023 | Adriana Matos | | Draft Release |
| **1** | 9/26/2023 | Courtney Eaton | | Submitted Version |

# Table of Contents

# List of Tables

# List of Figures

# 1. Introduction

## *1.1 Purpose and Scope*

This specification defines the technical requirements for the development items and support subsystems delivered to the client for the project. Figure 1 shows a representative integration of the project in the proposed CONOPS. The verification requirements for the project are contained in a separate Verification and Validation Plan.

The Crypto-analysis Resistant Digital Key FOB (C-kF) system will provide a secure and efficient solution for managing vehicle access and control. This system includes an Android application, encryption and communication subsystem, and hardware components acting as the vehicle. The Android application will facilitate user and hardware registration by capturing a signature through a gesture or stylus input coupled to a random number generated (associated with hardware - mobile device) and manage communication with the vehicle's Electronic Control Units (ECUs) to send commands for vehicle unlocking and starting mechanism. The hardware subsystem includes a microprocessor, LEDs, CAN bus network, and Bluetooth connectivity. This will provide seamless hardware-software integration. For initial device/key FOB registration, the communication and cryptography subsystem will provide security through the utilization of the Diffie-Hellman Key Exchange and AES-256 algorithm. This ensures the system is impervious to remote crypto-analysis attacks. Once the mobile device and ECUs are “coupled/registered” to each other, the communication and cryptography subsystem can ensure the system is resistant to crypto-analysis attacks through the utilization of SHA-256 in sending the locking/unlocking and starting engine command. In essence, this system's scope encompasses the transportation industry's need for a secure and user-friendly digital key FOB solution.

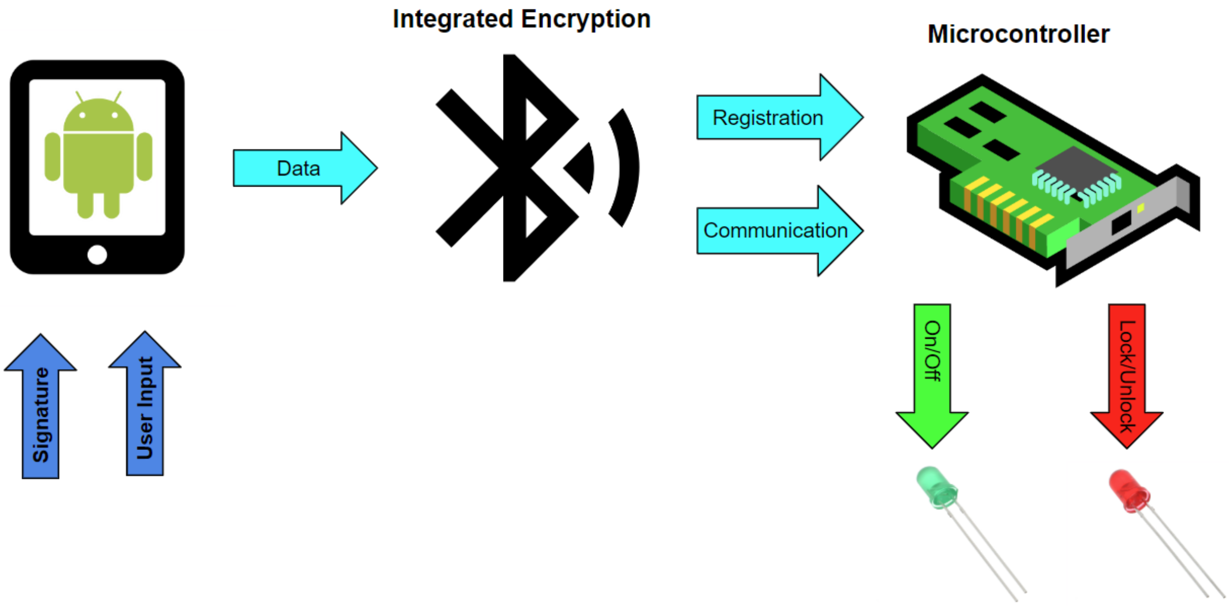

**Figure 1. C-kF Conceptual Image**

### *1.2 Responsibility and Change Authority*

The team leader, Adriana Matos-Almodovar, is responsible for making sure the project's requirements are met. If any change is made to the project, whether it be the deliverables or final demonstration, it must be approved by the team leader, Adriana Matos-Almodovar, as well as the team's sponsor, Peter Choi.

| **Subsystem** | **Responsible Team Member** |
|---|---|
| Locking Mechanism-Hardware | Adriana Matos |
| Android Application Development | Jeremy Hein |
| Communication and Cryptography | Courtney Eaton |

**Table 1. Subsystem Responsibilities**

## 2. Applicable and Reference Documents

### *2.1 Applicable Documents*

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

| **Document Number** | **Revision/Release Date** | **Document Title** |
|---|---|---|
| IEEE 802.15.1 | 2002 | IEEE Standard for Telecommunications and Information Exchange Between Systems- LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) |
| IEEE 1363 | 2000 | IEEE Standard Specifications for Public-Key Cryptography |
| IEEE 1363.3 | 2013 | IEEE Standard for Identity-Based Cryptographic Techniques using Pairings |
| IEEE 287.1 | 2021 | IEEE Standard for Precision Coaxial Connectors at RF, Microwave, and Millimeter-Wave Frequencies – Part 1: General Requirements, Definitions, and Detailed Specifications |
| FIPS PUB 180-4 | August 2015 | Secure Hash Standard (SHS) |
| Processing Standard Publication 197 | November 26, 2001 | Announcing the Advanced Encryption Standard (AES) |

**Table 2. Applicable Documents**

### *2.2 Reference Documents*

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

| Document Number | Revision/Release Date | Document Title |
|---|---|---|
| ISO 11898-1 | 2015 | Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signaling |

**Table 3. Reference Documents**

### *2.3 Order of Precedence*

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings, or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

## 3. Requirements

### *3.1 System Definition*

The C-kF will be comprised of three subsystems: the Android Application Development Subsystem, the Communication and Cryptography Subsystem, and the Locking Mechanism Subsystem. The Android Application Development subsystem is responsible for creating an Android application that will act as a digital key FOB and is able to communicate with the Locking Mechanism. This will be done by providing an interface for any user to easily be able to send the wanted command to their vehicle. The Communication and Cryptography subsystem is responsible for the Registration and Communication processes which will be implemented on both the Android application and the locking mechanism to securely send information between the mobile device and the vehicle. Finally, the Locking Mechanism subsystem will take the place of the vehicle as a way to demonstrate that the application is successful in sending the commands. It is responsible for mimicking a vehicle’s ECU’s internal communication system to “lock/unlock” or “turn on/off” the vehicle using the digital key FOB Android application. Together these subsystems will work to provide the user with an application to securely access and start their vehicle thus preventing any relay, man-in-the-middle, or other attacks to gain unwanted access to the user’s vehicle.

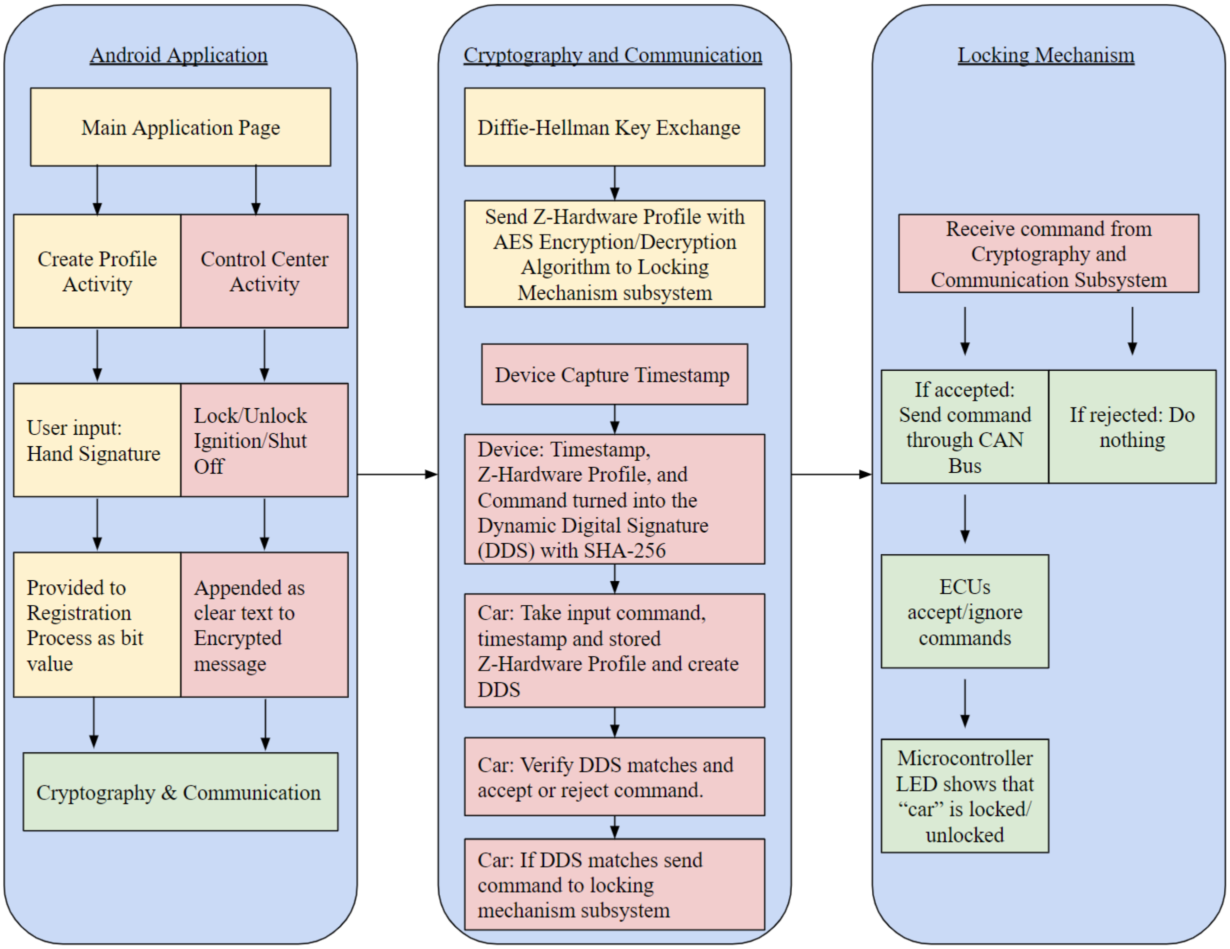
Android Application
Main Application Page
Create Profile Activity
Control Center Activity
User input: Hand Signature
Lock/Unlock Ignition/Shut Off
Provided to Registration Process as bit value
Appended as clear text to Encrypted message
Cryptography & Communication
Cryptography and Communication
Diffie-Hellman Key Exchange
Send Z-Hardware Profile with AES Encryption/Decryption Algorithm to Locking Mechanism subsystem
Device Capture Timestamp
Device: Timestamp, Z-Hardware Profile, and Command turned into the Dynamic Digital Signature (DDS) with SHA-256
Car: Take input command, timestamp and stored Z-Hardware Profile and create DDS
Car: Verify DDS matches and accept or reject command.
Car: If DDS matches send command to locking mechanism subsystem
Locking Mechanism
Receive command from Cryptography and Communication Subsystem
If accepted: Send command through CAN Bus
If rejected: Do nothing
ECUs accept/ignore commands
Microcontroller LED shows that "car" is locked/ unlocked

**Figure 2. Block Diagram of System**

As shown in Figure 2 above, the Android Application Development subsystem will be connected to the Communication and Cryptography subsystem, and the Communication and Cryptography subsystem will be connected to the Locking Mechanism subsystem.

During the Registration Process (shown by the yellow boxes above), the Android application subsystem will show a screen to the user and allow them to enter their information, vehicle information, and a hand signature done with their finger or a stylus. This image will then be handed over to the Communication and Cryptography subsystem. This subsystem will convert the hand signature into the Z-Hardware Profile and with the symmetric shared encryption key (AES-256) be securely sent to the Locking Mechanism subsystem where it will be stored.

During the secure communication process (shown by the red boxes above), the user will select a command through the application. The Communication and Cryptography subsystem will then have the device capture the unix-time and the Z-Hardware profile will be used to modify the captured unix-time and, using SHA-256, be turned into the Dynamic Digital Signature (DDS). This signature will then be sent over to the vehicle along with the timestamp and command in clear text. The vehicle will then use the shared Z-Hardware profile and SHA-256 to create its own DDS. The mobile device's DDS will then be verified against the vehicle's DDS. If both DDSs match then the command will be accepted and sent to the Locking Mechanism subsystem. If they do not match, then the command will simply be ignored.

## *3.2 Characteristics*

### 3.2.1 Functional / Performance Requirements

### 3.2.1.1 Android Application Requirements

#### 3.2.1.1.1 Profile Creation/Hand Signature

When creating a vehicle Profile, the user must input a hand signature using either their finger or a stylus. This signature shall be converted to a bit value and mixed with a pseudo-random number to be used as a common Z-hardware profile for the mobile device and ECUs.

> *Rationale: This requirement is necessary to provide a non-replicable dynamic value to the Communication and Cryptography subsystem.*

#### 3.2.1.1.2 Multiple Profiles

The user shall be able to create multiple profiles for different vehicles/ECUs and select the profile they want to control from the main page.

> *Rationale: Multiple profiles allow the user to control multiple vehicles with one device which is useful as a large majority of the population uses more than one vehicle.*

#### 3.2.1.1.3 Control Panel

After selecting the desired Vehicle Profile, the user will be navigated to a control panel screen and have 4 options for controlling the vehicle: Lock, Unlock, Ignition, and Shut-off. The application page shall also indicate the pre-existing condition of the vehicle showing if it is On/Off or Locked/Unlocked aiding the user in knowing which command to send.

> *Rationale: The Control Panel is necessary to provide the user with a way of inputting their commands and relaying them to the Communication subsystem.*

#### 3.2.1.1.4 FAQ Page

Each page in the application will include a button that leads to the FAQ page that will include basic instructions and guides for operating the software.

> *Rationale: Providing the user with basic operating information is an ease-of-use improvement aimed at assisting people with less knowledge of operating digital software and devices and providing users with the ability to troubleshoot the most likely problems they may encounter.*

#### 3.2.1.1.5 Android OS Version

The application will run on Android 10. The app shall have a minimum Android SDK of 29. The compiled SDK will be 31. This Android OS runs on approximately 75% of all Android devices according to Android Studio.

*Rationale: An Android application requires an OS to run the software. Android 10 was chosen as it includes modern features and still provides widespread support for older devices.*

#### 3.2.1.1.6 Status Messages

Error messages returned by the Communication and Cryptography system will be shown to the user such as “Registration Failure” and “Connection Error”. The inverse of those messages will be displayed for successful operations. Examples are “Registration Successful” and “Connection Successful”.

*Rationale: It is necessary to inform the user of the success or failure of their attempted operation for them to have no doubt that the application is performing as intended.*

### 3.2.1.2 Communication and Cryptography Subsystem

#### 3.2.1.2.1 Securely Send/Receive Data

The Android application and locking mechanism should be able to send/receive data securely to avoid man-in-the-middle attacks made to one’s vehicle. To accomplish this, the Diffie-Hellman Key Exchange Protocol shall be used to exchange a one time encryption key between the device and vehicle. The AES-256 Encryption/Decryption Algorithm shall be used to send the Z-Hardware Profile from the device to the vehicle. Finally, SHA-256 shall be used to create the Dynamic Digital Signature for securely sending commands to the vehicle.

*Rationale: This requirement is to ensure that the communication between the vehicle and device is secured using the most widely accepted encryption/decryption and hashing algorithms.*

### 3.2.1.3 Locking Mechanism Subsystem

#### 3.2.1.3.1 Accurate Demonstration of Lock/Unlock and Start/Stop

The Locking Mechanism subsystem shall accurately indicate the “vehicle” is locked/unlocked using the LED lights on the PCB. It will also indicate if the “vehicle” is turned on/off through the use of LED lights.

*Rationale: This is a basic requirement to show that the demonstration PCB is working properly and the secure communication between the Android application and the Locking Mechanism is successful.*

#### 3.2.1.3.2 Bluetooth Capabilities

The Locking Mechanism subsystem shall have bluetooth capabilities in compliance with the IEEE 802.11ac standards.

*Rationale: This is a basic requirement to allow for communication between the Android application and the Locking Mechanism.*

### 3.2.2. Physical Characteristics

#### 3.2.2.1. Structural

The Locking Mechanism will consist of a custom made PCB Microcontroller inside of a housing unit for protection.

*Rationale: This is a requirement from the sponsor to mimic a vehicle and show that a digital key FOB can have secure communication with that vehicle.*

### 3.2.3. Electrical Characteristics

#### 3.2.3.1. Inputs

The input into the system shall not cause any damage or malfunction to the main internal communication system (CAN Bus). No man-in-the-middle input attacks should affect the system whatsoever as any data inputs will be confirmed by the Communication and Cryptography subsystem to solely come from the Android application being developed. Along with data input from the application, the mechanism shall also intake power from a wall outlet adapter.

*Rationale: Due to the design of the microcontroller being used, the chance of a malfunction due to power failure should be limited.*

#### 3.2.3.1.1 Input Voltage Level

The input voltage level for the Locking Mechanism Microcontroller will be about +7V to +12V of DC voltage coming from the power supply adapter. The power shall come from a power supply adapter that is connected to an AC wall socket.

*Rationale: An AC wall socket will allow for ease of use of the Locking Mechanism as wall outlets are fairly common. The microcontroller will be able to accept a minimum of 6V and a maximum of 20V using the Jack connector so a input voltage level of 7V to 12V was chosen to provide a buffer in case of power spikes.*

#### 3.2.3.1.2 External Commands

The Locking Mechanism shall take in and store the external commands (Z-Hardware Profile) that come from the Android application.

*Rationale: This allows for the Z-Hardware profile to be used to verify the identity of the device attempting to give a command to the vehicle.*

#### 3.2.3.2. Outputs

#### 3.2.3.2.1 Data Output

The Locking Mechanism's output shall be a visible representation that the system is either locked, unlocked, turned on, or turned off. The locking/unlocking representation will be done by using a red LED: when the LED is on it signifies that the system is "locked" and when the LED is off it signifies that the system is "unlocked". The turned on/off representation will be done by using a green LED: when the system is “on” the LED will turn on and vice versa.

> *Rationale: This allows for the command passed on from the application to be visibly shown on the Locking Mechanism for demonstration purposes.*

#### 3.2.3.3. Connectors

The Locking Mechanism Subsystem shall follow the International Electrotechnical Commission IEC 60130-10:1971 standard for electrical connectors.

> *Rationale: Conform to the standard for coaxial connectors.*

#### 3.2.3.4. Wiring

The Locking Mechanism Subsystem shall reference the ISO 11898-1 standard for CAN bus architecture wiring.

> *Rationale: Conform to wiring standard.*

### 3.2.4. Environmental Requirements

The C-kF system does not have any environmental requirements.

### 3.2.5. Failure Propagation

The C-kF system should not allow any errors or communication interruptions of any sort. For project simplicity, a couple of security assumptions are made: 1. hackers do NOT have physical access to the ECUs, and 2. hackers do NOT have control over the mobile device.

#### 3.2.5.1. Wireless Communication Error

The C-kF system will utilize its Communication and Cryptography subsystem to identify any bluetooth error or Man-in-the-Middle attack between the Android application and the Locking Mechanism.

> *Rationale*: *This will ensure the digital key FOB is crypto-analysis resistant as well as helping to cover any possible bluetooth error.*

#### 3.2.5.2. Cryptography Error

The Communication and Cryptography subsystem will send a signal to the Android application subsystem if any error occurs while encrypting/decrypting the sent signal.

> *Rationale: This will allow the user of the application to resend a signal or show that the Communication subsystem caught a potential attack.*

#### 3.2.5.3. Electrical Errors

The Locking Mechanism has a built-in green LED light to show that the system is fully running as it should and that the PCB is connected to power. If the green LED is not on, a problem has occurred in the subsystem and a reset button will be available to reset the power.

*Rationale: For demonstration purposes, this will allow the user to reset power and restart the system if any electrical errors occur.*

# 4. Support Requirements

The Android Application will include a FAQ page with basic instructional and operational information allowing users who do not have previous experience with similar apps to navigate and use the interface and controls with ease. There is no warranty for this software as we are not providing a new device but simply installing additional software to an already existing digital infrastructure. If a large company was providing this service a technical support service would be necessary but this project's design scope does not include this solution. When downloading the application from the Google Play Store, the customer will be shown the device requirements for running the application and whether their device meets them.

# Appendix A: Acronyms and Abbreviations

Below is a list of common acronyms and abbreviations used in this project.

| | |
|---|---|
| AES | Advanced Encryption Standard |
| CAN | Controlled Area Network |
| C-kF | Crypto-analysis Resistant Digital Key FOB |
| DDS | Dynamic Digital Signature |
| ECU | Electronic Control Unit |
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| KMS | Key Management Service |
| LED | Light-Emitting Diode |
| OTA | Over-The-Air |
| PCB | Printed Circuit Board |
| PKI | Public Key Infrastructure |
| SDK | Software Development Kit |
| SHA | Secure Hash Algorithm |
| V | Volt |
| ZAV | Zero-Trust Application for Vehicle |
| ZTA | Zero Trust Application |

# Appendix B: Definition of Terms

Dynamic Digital Signature - The resulting ciphertext created when the Z-Hardware Profile, and unix-time are put through the SHA-256 Algorithm.

Z-Hardware Profile - Unique combination of the hand imputed signature (with a finger or stylus) and a pseudo-random number which is used to verify the identity of the device trying to send commands to the vehicle.

# Crypto-analysis Resistant Digital Key FOB

Adriana Matos
Courtney Eaton
Jeremy Hein

# INTERFACE CONTROL DOCUMENT

REVISION – 1
September 26, 2023

# Interface Control Document

for

# Cypto-analysis Resistant Digital Key FOB

Prepared by:

Project Team 9/26/2023

Approved by:

Adriana Matos 9/26/2023

John Lusher II, P.E. Date

T/A Date

## Change Record

| Rev. | Date | Originator | Approvals | Description |
|---|---|---|---|---|
| - | 9/23/2023 | Adriana Matos | | Draft Release |
| 1 | 9/26/2023 | Courtney Eaton | | Submitted Version |

# Table of Contents

# 1. Overview

This document will provide more detail on the interface between the user, Android application on the device and locking mechanism hardware. It will first describe the physical, thermal, and electrical interfaces of the device and locking mechanism. After, the communication interface between the device and locking mechanism will be described.

# 2. References and Definitions

## *2.1 References*

Refer to section 2.1 and 2.2 of the Functional System Requirements Document.

## *2.2 Definitions*

| | |
|---|---|
| AES | Advanced Encryption Standard |
| CAN | Controlled Area Network |
| C-kF | Crypto-analysis Resistant Digital Key FOB |
| DDS | Dynamic Digital Signature |
| ECU | Electronic Control Unit |
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| KMS | Key Management Service |
| LED | Light-Emitting Diode |
| mm | Millimeter |
| OTA | Over-The-Air |
| PCB | Printed Circuit Board |
| PKI | Public Key Infrastructure |
| SDK | Software Development Kit |
| SHA | Secure Hash Algorithm |
| V | Volt |
| ZAV | Zero-Trust Application for Vehicle |
| ZTA | Zero Trust Application |

# 3. Physical Interface

## *3.1 Weight*

The weight of the PCB locking mechanism will be about 25 grams in total, not including the housing unit. During the demonstration process a tablet will be used to demonstrate the application's capabilities and the weight of the tablet is determined by the Android device's manufacturer.

## *3.2 Dimensions*

The dimensions of the PCB containing the locking mechanism will be about 68 mm x 53 mm. This is based on different microcontrollers that have similar capabilities and characteristics.

# 4. Thermal Interface

The microcontroller does not need a heat sink and should be able to dissipate the heat created on its own.

# 5. Electrical Interface

## *5.1 Primary Input Power*

The Locking Mechanism will be powered by an 120V AC wall outlet. Though the microcontroller does not need to work with 120V AC power, an adapter will be used to provide the microcontroller with at least the minimum voltage needed to function.

## *5.2 Signal Interfaces*

The Locking Mechanism will have an internal communication network that allows Electronic Control Units (ECUs) to send internal signals that allow other ECUs to accept or ignore. External signals will be sent over from the user control interface and broadcasted through the microcontroller's internal communication network.

## *5.3 User Control Interface*

The user control interface is the Android device application that acts as the Locking Mechanism's digital key FOB. The user's input signal will be encrypted and sent over to the microcontroller to be decrypted and accepted by the electronic control units within the microcontroller.

# 6. Communications/Device Interface Protocols

## *6.1 Bluetooth Communication*

The microcontroller will have an ESP32 chip assembled with it that has bluetooth capabilities using the IEEE 802.11ac standards. This bluetooth connection will be used so the Android application can communicate with the locking mechanism.

## *6.2 Diffie-Hellman Key Exchange Protocol*

The Diffie-Hellman Key Exchange Protocol will be used during the registration process in order to create and set up the symmetric key to be used by the device and vehicle during the transfer of the Z-Hardware Profile.

## *6.3 Secure Hash Algorithm (SHA), 256 bits*

SHA-256 will be used to create the Dynamic Digital Signature from the Z-Hardware Profile and unix-time. It will operate based on the Secure Hash Standard set in place by the National Institute of Standards and Technology (NIST).

## *6.4 Advanced Encryption Standard (AES), 256 bits*

The AES-256 algorithm will be used to securely send the Z-Hardware Profile from the device to the vehicle. It will operate based on the specifications put in place by the NIST for this standard.

# Crypto-analysis Resistant Digital Key FOB

Adriana Matos
Courtney Eaton
Jeremy Hein

# Schedule and Validation Plan

REVISION – 1
September 26, 2023

## Schedule

Legend: Not Started In Progress Completed Behind Schedule

| Week of: | Jeremy | Adriana | Courtney |
|---|---|---|---|
| August 21st | Team Assigned | | |
| August 28th | Assign Subsystems | | |
| September 4th | Android application research | Microcontroller Research | Cryptography Research |
| September 11th | Explore Android Studio and set up project space | Create an overview list of potential parts needed | Finalize Registration and Communication Processes with Sponsor |
| September 18th | Create structure for App activities and start building intent value sharing in between activities | Order first round of parts for Microcontroller | Plan code for subsystem |
| September 25th | Design Application look and theme | Begin designing Microcontroller on Altium | Create code for the creation of the Z-Hardware Profile from a hand signature image |
| October 2nd | Code profile selection and creation | Begin to breadboard design and debug | Finish creation of Z-Hardware Profile from hand signature image |
| October 9th | Code Error/Success Messages | Continue Prototype/Debug | Code Diffie-Hellman Key Exchange Protocol |
| October 16th | Code Hand signature conversion to bit value | Continue Prototype/Debug | Code AES Encryption Process |
| October 23rd | Add Bluetooth capabilities to app | Finish Prototype | Code AES Decryption Process |
| October 30th | Code intent value sharing for registration process | Design PCB on Altium and order parts | Code SHA Message Digest Process |
| November 6th | Buffer Week | Buffer week | Code Vehicle |

| | | | |
|---|---|---|---|
| | | | verification of Digital Dynamic Signature |
| November 13th | Buffer Week | Solder parts onto PCB | Buffer Week |
| November 20th | Final Touches | Test/Finalize PCB | Final Touches |
| November 27th | Final Demonstration | | |

## Validation Plan

### Android Application Subsystem - Jeremy

| Test Parameter: | Expected Result: | Date Tested: |
|---|---|---|
| App Functionality | Application will start on device running a compatible OS | 9/19/2023 |
| App Navigation | Users will be able to freely navigate app without getting stuck on any page and with no errors | 9/26/2023 |
| Intent Value Sharing | Activities will share necessary information when certain fields are filled or buttons pressed | |
| Application Design | Application will be viewable as a product with modern designs and layouts | |
| Hand signature | During registration process, The application will take in a user gesture and convert it to an image | |
| Registration Procedure | User will create a Car Profile and it app will store the profile parameters | |
| Communication Procedure | Application will relay user commands to the Communication subsystem | |
| FAQ page | User will be able to enter FAQ page from starting screen | |

| | | |
|---|---|---|
| Bluetooth capability | Application will be able to connect to Bluetooth devices | |

Locking Mechanism (Hardware) Subsystem - Adriana

| Test Parameter: | Expected Result: | Date Tested: |
|---|---|---|
| Red/Green LED Test | Will turn on when a specific voltage is applied & will turn off when opposite is applied | |
| ESP32 Chip/CAN Transceiver | CAN Bus Network sends and receives information | |
| Power | Microcontroller is able to turn on/off with no problems or excessive heat dissipation | |
| Overall Microcontroller | Whole system works together and no interrupted current paths | |
| Bluetooth | Microcontroller is able to accept data over Bluetooth interface | |

Communication & Cryptography Subsystem - Courtney

| Component: | Test: | Date Tested: |
|---|---|---|
| Z-Hardware Profile | Hand signature image turns into a unique byte number | |
| | Signature byte number is combined with a pseudo-random number | |
| Diffie-Hellman Key Exchange Protocol | Device and vehicle are able to create their own pseudo-random number combined with the public starting key | |
| | Device and vehicle create AES symmetric key from the other’s sent information | |

| | | |
|---|---|---|
| AES Encryption Code | Z-Hardware Profile is encrypted | |
| AES Decryption Code | Z-Hardware Profile is decrypted | |
| SHA Algorithm Code | Unix-time is captured | |
| | Z-Hardware Profile and unix-time are created into the Dynamic Digital Signature for the device | |
| | Z-Hardware Profile and unix-time are created into the Dynamic Digital Signature for the vehicle | |
| Verification Code | Vehicle is accurately able to verify if Dynamic Digital Signature created by the device matches that created by the car | |

| Communication & Cryptography | Milestone | Comments |
|---|---|---|
| September 19th | Research regarding how SHA, AES, & Diffie-Hellman Work Completed | |
| September 26th | Diffie-Hellman Key Exchange Planning Completed | |
| October 3rd | Diffie-Hellman Key Exchange Coded and Working | |
| October 10th | SHA Code Planning Complete | |
| October 17th | | |
| October 24th | SHA Coded and Working | |
| November 7th | AES Code Planning Complete | |
| Nobember 14th | | |
| November 21st | AES Coded and Working | |
| November 29th | Final Demostration | |

| Android Application | Milestone | Comments |
|---|---|---|
| September 19th | Create UI template design | |
| September 26th | Impliment UI design with buttons/ Study kotlin | |
| October 3rd | Create profiles for application | |
| October 10th | SHA Code Planning Complete | |
| October 17th | | |
| October 24th | SHA Coded and Working | |
| November 7th | AES Code Planning Complete | |
| Nobember 14th | | |
| November 21st | AES Coded and Working | |
| November 29th | Final Demostration | |

| Locking Mechanism | Milestone | Comments |
|---|---|---|
| September 19th | Send ECEN Business Office first round of product order | |
| September 26th | Microcontroller Research and start design | |
| October 3rd | Finish designing Microcontroller | |
| October 10th | Begin prototype | |
| October 17th | | |
| October 24th | Protoype done | |
| November 7th | Design PCB on Altium and order | |
| Nobember 14th | Solder parts onto PCB | |
| November 21st | Test PCB | |
| November 29th | Final Demostration | |