University of Colorado Colorado Springs

Design Document

eCTF MITRE 2023

Version 1.1

TEAM UCCS1 “RGB”
3-3-2023

Table of Contents

Protected Automotive Remote Entry Device (PARED) Design

1 Purpose

1. Designing of secure firmware for Protected Automotive Remote Entry Device (PARED).
2. Designing for the protection against an adversary having physical access to the fob and car.
3. Mitigating the constraints on the functional and non-functional requirements by the security requirements.
4. Testing the PARED design for the security goals implementation and mitigation of the impacts and constraints.
5. Designing the Build Environment and Build Tools to meet the security goals.

2 Assets

Asset Number	Asset	Category
HT.PF	Packaged Feature Procedure	Host Tool
HT.EF	Enabled Feature Procedure	Host Tool
HT.PAIR	Pair Fob Procedure	Host Tool
PFOB	Paired Fob	Fob
UFOB	Unpaired Fob	Fob
CAR	Car Firmware Binary	Car
CAR.ID	CAR ID	Car
CAR.ROM	Car EEPROM Data	Car
CAR.UL	Unlock Car Function	Car
CAR.UL.MSG	Unlock Car Message	Car
CAR.ST	Start Car Function	Car
PFOB.BIN	Paired Fob Firmware Binary	Fob
PFOB.ROM	Paired Fob EEPROM Data	Fob
UFOB.BIN	Unpaired Fob Firmware Binary	Fob
UFOB.ROM	Unpaired Fob EEPROM Data	Fob
(P/U)FOB.F#	Feature Number Data	Fob
(P/U)FOB.F.MSG	Feature Message	Fob
HST.SEC	Host Secrets	Inputs to Host Tools
PPIN	Pairing PIN	Inputs to Host Tools

3 Security Goals

There are six security goals that need to be met by this system which are as follows:

SG1: A car should only unlock and start when the user has an authentic fob that is paired with the car.

SR1.1. The system shall FR3.4 (Unlock and Start Car) only by PFOB (paired fob).
Note: SR1.1 and SR4.1 are codependent.

To achieve SG1, SR1.1, SR2.1 and SR4.1, need to be implemented. Based on assumption AS6 that paired fob is authentic. Implementing SR2.1 (prevent the duplication) and SR4.1 (establish the unique link between PFOB, PPIN and CAR) shall have application of Integrity – one of the security triad CIA, ensures the PFOB (paired fob) is only able to complete FR3.4.
It puts constraints on FR1.4, FR1.5, FR3.3, FR3.4, and FR5.

SG2: Revoking an attacker's physical access to a fob should also revoke their ability to unlock the associated CAR

SR2.1. The System shall prevent reading of the PFOB (paired fob). This includes FOB (fob firmware binary) and FOB.ROM (fob EEPROM Data). Assume, the PFOB with CAR goes for valet parking and unknown guy has access to the PFOB. Who is attacker and tried to copy PFOB. Once the PFOB and CAR is retrieved from valet parking, Copied fob (unauthentic) shall not have access to the CAR

To achieve this goal, SR2.1 restricts the cloning of the FOB and FOB.ROM by restricting its read access (application of Confidentiality).
It has constraints on FR1.2, FR1.5, FR5 and FR6.

SG3: Observing the communications between a fob and a car while unlocking should not allow an attacker to unlock the car in the future.

SR3.1. The PFOB (Paired Fob) shall not leak the PPIN (Pairing PIN). Constrains FR1 and FR3.3.
SR3.2. Altering the signal shall not allow FR3.4 (access to unlock the car).
SR3.3. Replaying the signal shall not allow FR3.4 (access to unlock the car).

To achieve this goal, SR3, SR4.4, SR4.5 shall be implemented with signals encryption during the communication, having each signal blocks creation with timestamp, sequence, and checksum information. Intention is to apply Confidentiality and Integrity – two of the security triad CIA.
It has constraints on FR1.2, FR1.5, FR5 and FR6.

SG4: Having an UFOB (Unpaired Fob) should not allow an attacker to unlock a car without a corresponding PFOB (Paired Fob) and PPIN (Pairing PIN)

SR4.1. The system shall provide PFOB (paired fob) and PPIN (pairing PIN) uniquely for each CAR.
Note: SR1.1 and SR4.1 are codependent.

To achieve this goal, SR4 shall reject the communication from UFOB, by detecting and analyzing the signal blocks by application of Integrity – one of the security triad CIA and principle of least privileges such that the UFOB will not have any access to communicate with the CAR.
It has constraints on FR3.4 and FR5.

SG5: The car owner shall not be able to add new features to a fob that did not get packaged by the manufacturer.

- SR5.1. No one other than the manufacturer shall package new features to a fob.
- SR5.2. The system shall not allow tampering with the packaged feature
- SR5.3. The system shall prevent an owner from adding an unpackaged feature.
- SR5.4. The system shall prevent an attacker from escalating privileges on a feature-limited fob.

To achieve this goal, SR5 shall implement signature verification and checksum verification in HT.PF, HT.EF and PFOB. It applies Integrity – one of the Security triad CIA.
It has constraints on FR3.2, FR3.4.1 and FR5.

SG6: The system shall not allow HT.PF (packaged feature) of one car to enable the same feature on another car.

- SR6.1. When a car owner paid for an upgraded feature and received an upgrade from the manufacturer then the Car owner shall not be allowed to upgrade it on the other car. (In a situation, car owner has multiple cars from same manufacturer and request and pay for the feature on only one CAR.ID using the car shared key)
- SR6.2. An attacker shall not be allowed to use feature upgrade when he has paid for an upgraded feature and the manufacturer sends them using other car details

To achieve this goal, SR6 shall be implemented in HT.PF by applying Integrity – one of the security triads. HT.PF shall verify the Car Details of the packaged feature from the manufacturer and the Car Details in HT.PF, before proceeding with the feature packaging to the PFOB.
It has constraints on FR1.5, FR3.1, FR3.2 and FR3.4.

4 Functional Requirements

The functional requirements (labeled as FRx – x corresponds to numbers 1, 2, …. 1.1, 1.2….) are implemented in the example design code provided by MITRE and are listed below (FR4). The final implementation shall not break them.

FR1. Build PARED System: This is intended for the Car Manufacturer's Secure Facility. Build PARED System produces the following assets/tools:

- FR1.1. Host computer Docker Image: Pre-built image for host computer.
- FR1.2. FR1.1 shall have all Host Tools and their dependencies like Host secrets and Pairing PIN
- FR1.3. Host Secrets File shall be created and available in FR1.1

FR1.4. Car Binary and EEPROM File.
FR1.5. Paired Fob Binary and EEPROM File.
FR1.6. Unpaired Fob Binary and EEPROM File.

FR2. Load Devices: The FR1.4, FR1.5 and FR1.6 (firmware and EEPROM contents) shall be loaded onto the microcontrollers. No modifications of any part in this activity.

FR3. Host Tools: These are intended for the Car Owner and produce the following assets/requirements.
FR3.1. Package Feature: into the paired Fob. Inputs to this tool are Car ID, Host Secrets and Feature Number.
FR3.2. Enable Feature: Adds the packaged feature to the paired fob. After this activity, when the fob is used to unlock and start the car, the Car will recognize the added feature.
FR3.3. Pair Fob: Pair UFOB (Unpaired Fob) with FR1.4 (a car). Input to this tool is Pairing Pin only. But require the connection with unpaired and a paired fob as mentioned in *Figure 6, section 3.3.2 of 2023 eCTF Rules v1.0.pdf.*
FR3.4. Unlock and Start Car: SW1 user button press on paired fob initiates the communication with car device to unlock and start the car.
FR3.4.1. Unlock: Car shall print the unlock message and the list of all enabled features.

FR4. The directory structure of the design shall match with the below:
FR4.1. eCTF Tools: https://github.com/mitre-cyber-academy/2023-ectf-tools
FR4.2. Example Design: https://github.com/mitre-cyber-academy/2023-ectf-insecure-example

FR5. The PARED system shall comply with the following size restrictions:

COMPONENT	**SIZE**
PPIN (Pairing PIN)	6 hexadecimal digits
FOB.F# (Feature Number)	8-bit unsigned integer
FOB.F.MSG (Feature Message)	Max 64 bytes
CAR.UL.MSG (Unlock Message)	Max 64 bytes
CAR.ID (Car ID)	8-bit unsigned integer

FR6. Memory Requirements
FR6.1. PARED system shall compile any firmware (car and fob) such that its first instruction is at address 0x00008000.
FR6.2. The PARED system shall not use any Flash memory located below 0x00008000 address.
FR6.3. The last 256 bytes of EEPROM shall be reserved for messages that get printed over UART when the car is unlocked and when specific features are enabled.
FR6.4. PARED system may not permanently commit Flash memory write protections.
FR6.5. If PARED system uses interrupts, they shall place a copy of the vector table in SRAM or Flash after the start of their system.

5 Non-Functional Requirements

This section would list the non-functional requirements which neither fall under Functional nor the Security Requirements as follows:

NFR1. System Design Acceptance Criteria
- NFR1.1. Submission of all source code and all documents.
- NFR1.2. Pass all Functional Requirements.
- NFR1.3. Follow all Rules.

NFR2. All the Source Code and Documents of Accepted Designs are distributed as Attack Phase artifacts.

NFR3. The PARED system shall comply with the following memory size restrictions:

COMPONENT	**SIZE**
CAR (Car Firmware)	Max 110 KB
CAR.ROM (Car EEPROM Data)	Max 1792 bytes
FOB (Fob Firmware)	Max 110 KB
FOB.ROM (Fob EEPROM Data)	Max 1792 bytes

NFR4. The PARED system shall comply with the following timing restrictions:

OPERATION	**MAXIMUM TIME FOR COMPLETION**
Boot	1 Second
Pair Fob	1 Second
Package Feature	1 Second
Package Feature	1 Second
Unlock Car	1 Second

6 Build Tools and Environment Design

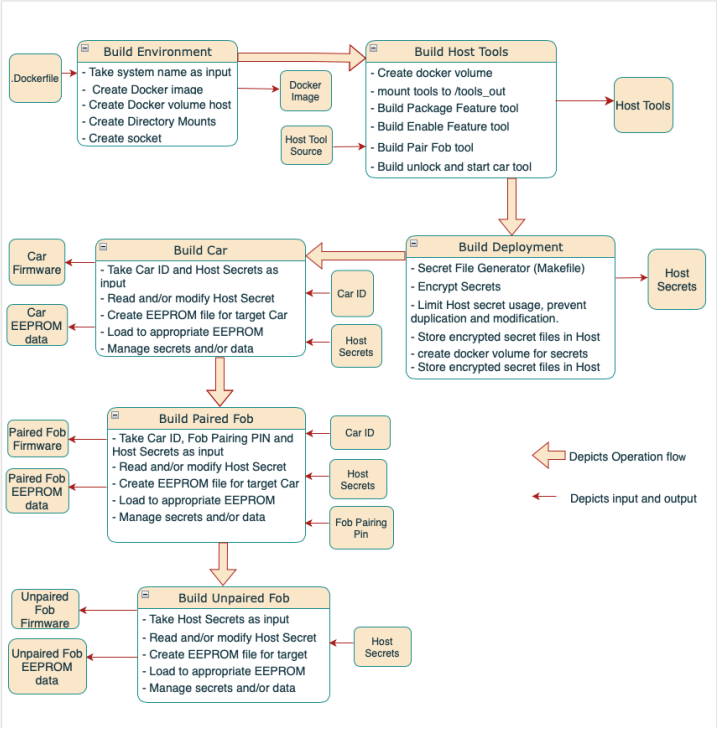

Figure 1

7 Test Design

Testing will be carried out to test each one of the functional requirements, security requirements and non-functional requirements as per the *Section 3 Scope*. Also, during the testing the interdependencies and the impacted requirements would be tested explicitly to ensure that any security requirement does not break existing functionality except for the constraints agreed upon as per *section 2 Security Goals*.

T1. Static Analysis Testing (White Hat)

T1.1 Static Code Scanning will validate that there are no new patches, numerical errors, input validation, race conditions, path traversals, pointers, and references, and more through the implementation of the GitHub built-in CodeQL tool.

T1.1.1 Code scanning will include linting the code to validate there are no insecure coding errors which could lead to easy attack vectors for the attack team through IntelliSense Code Linter used in Visual Studio Code.

T1.2 Memory Requirement testing will validate that the system firmware complies with all requirements outlined in FR6 through manual analysis.

T1.3 Size restriction testing will validate that the message, IDs, and other assets meet the size requirements as outlined in FR5 through manual analysis.

T2. Dynamic Analysis Testing (Black Hat) and Interactive Application Security Testing (IAST)

T2.1 Communication security between assets will validate that there are no leaked secrets, pairing pin, firmware, binaries and more by capturing traffic between devices with Wireshark and manually examining the contents of the traffic.

T2.2 Timing testing will validate that the system executes within the required time through the use of unit testing, manual analysis and Wireshark.

T2.3 Fob security testing will validate that the fob can only be paired to one car at a time, installed features are only installed from the manufacturer, features can only be installed on the paired car, and features cannot be enabled on an unpaired car through manual analysis.

T2.4 Car security testing will validate that the car can only be paired to one fob at a time, installed features are only installed from the manufacturer, features can only be installed from the paired fob, and features cannot be installed from an unpaired car through manual analysis.

T3. Original Analysis and Software Competition Analysis (SCA Testing)

T3.1 Component and library security testing will validate that there are no vulnerabilities or patches in common and popular components, particularly open-source components used in the system using the built-in Dependabot tool on gitHub.

T4. Functionality validation will be assumed valid if the environment builds and runs correctly.

7.1 Tools:

7.1.1 CodeQL Analysis by GitHub

- Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, Ruby and Kotlin developers.

7.1.2 Fortify On Demand Scan by Micro Focus

- Fortify's comprehensive static code analysis (SAST) for 27+ languages for DevSecOps workflows to build secure software faster.

7.1.3 OpenScap

7.1.4 Wireshark

8 Threat Identification and Mapping to Security Requirements

Action -> Asset -> Harm

Threat Number	Threat Descriptions	Security Requirement Number
THR1	Spoofing paired fob signals enable an attacker to send signals to unlock and start the car.	SR1.1, SR4.1
THR2	Cloning of the paired fob firmware can enable an attacker to unlock or start the car.	SR2.1, SR4.1, SR6.1
THR3	Replaying paired fob signals enables an attacker to unlock, start, and pair fobs to the car.	SR3.1, SR3.2, SR3.3
THR4	Side-channel analysis of the car's computation enables an attacker to extract car secrets.	SR3.1, SR3.2, SR3.3
THR5	Pairing an unpaired fob without a paired fob enables an attacker to unlock, start, and pair fobs to the car.	SR1.1, SR4.1
THR6	Extracting a pairing pin from a paired fob enables an attacker to unlock, start, and pair fobs to the car.	SR1.1, SR2.1, SR4.1
THR7	Adding an unpackaged feature into the paired fob enables an escalation of privileges.	SR5.1
THR8	Reusing a packaged feature from one car for another car enables an escalation of privileges.	SR6.1, SR6.2

8.1 Justifications (Why SRs protect against Threats)

In language of Detection/Prevention/Recovery

- THR1 (Spoofing) Prevented by SR2.1, SR3.2, SR4.1. Preventing an attacker from reading a paired fob's binaries and secrets prevents them from mimicking a paired fob's functionality. If a spoofed/altered signal does not match that of a paired fob, the car will not unlock.
- THR2 (Cloning) Prevented by SR2.1. The System shall prevent reading of the paired fob. This includes reading firmware data and binaries.
- THR3 (Replaying) Prevented by SR3.3. Replaying of the signal does not allow unlocking of the car.
- THR4 Addressed under Assumptions AS7
- THR5 Prevented by SG4: SR4.1. The system shall not allow a pairing of an unpaired fob without a pairing PIN and corresponding paired fob.
- THR6 Prevented by SG2: SR2.1 by preventing reading of the paired fob, including its secrets
- THR7 Prevented by SG5: SR5.2, SR5.3 and SR5.4. The system shall not allow unpackaged features in addition to fob.
- THR8 Prevented by SR6.1 and SR6.2. The system shall prevent enabling using a packaged feature for a car on another car.

8.2 Security Mechanism

SM1 SR1.1 (Authentication) - System recognizes a host secret that is not the Pairing PIN in order to authenticate paired fob on every unlock/start transaction:

- HMAC Framework SHA-1
- A car verifies the message's integrity.

SM2 SR2.1 Encrypting the fob binaries prevents reading firmware and ROM to maintain confidentiality.

- Encrypt all secrets stored in the ROM

SM3 SR3.1 (Authentication of fob to car communication) The car will recognize a host secret (be a counter or timer) that is not the pairing PIN in order to authenticate the fob at the time of transmission.

- Both the car and fob use a symmetric encryption key which is pre-shared during the build operations.
- The shared symmetric key is used for encrypting each unlock message using tiny-AES.
- The car decrypts a fob's message using tiny-AES

SR3.2 Hash function to prevent the modification of signals

- Each fob message contains SHA-1 message digest.

SM4 SR4.1 Verification of a paired fob and a pairing PIN

- HMAC Framework

SM5 SR5.1 Verification of the manufacturer signature

- Pre-shared keys will be used to verify the manufactured packages

SR5.3 HT.PF shall authenticate the feature number in the package feature and the feature number in the Host Tool, preventing the addition if not matching.

- During the pairing operation, the paired fob will store the feature info containing the car ID and feature number. The paired fob will also be shared with a key to decrypt the XOR encrypted packaged feature by the Host Tools. While enabling a new feature, the paired fob will decrypt the packaged feature and compare the stored car ID with the car ID in the new packaged feature and enable the feature if the two IDs match.

SR5.4 [Same as SR5.3 Mechanism]

SM6 SR6.1 (Authentication of car ID) [Same as SR5.3 Mechanism]

SR6.2 [Same as SR6.1 Mechanism]

9 System Architecture Diagram

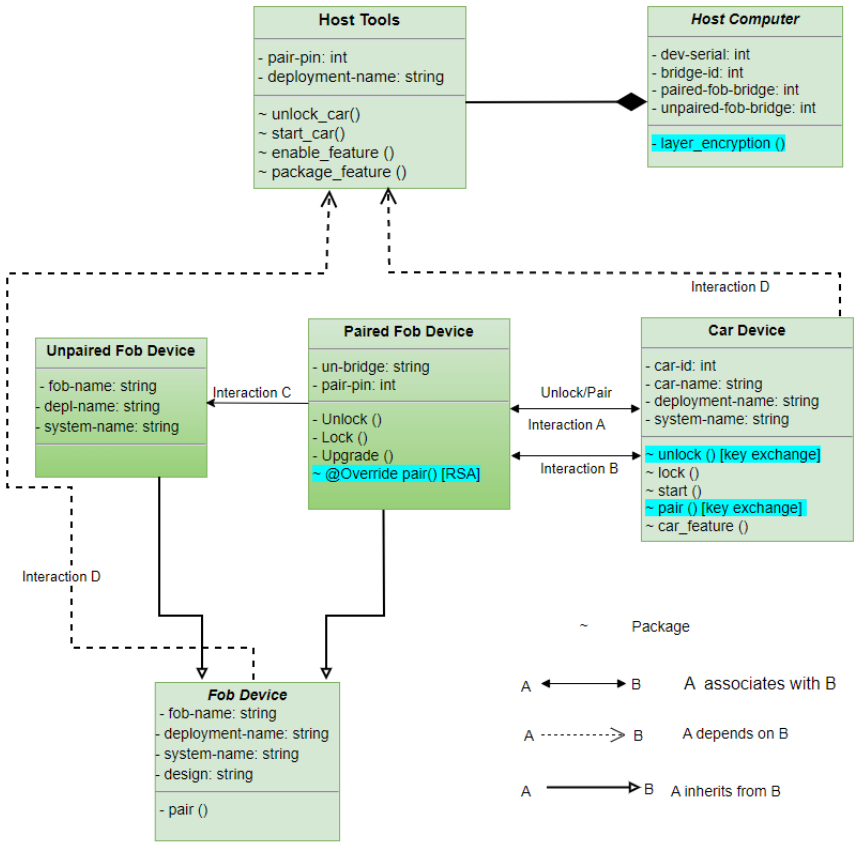

Figure 2

Figure 2 shows the architectural diagram of the system. In Host Computer, wherein the Host tools reside, SG2, layered encryption is implemented. The paired and unpaired fob devices are classes inherited from the parent class Fob Device. The highlighted methods depict the enforcement of security mechanisms.

The Interactions between components are as follows:

- Interaction A occurs between the paired fob and the car. The key exchange in correspondence to SG3 uses the Diffie-Hellman key exchange algorithm.
- Interaction B implements the security mechanism SG4, a digital signature using the RSA algorithm.
- Interaction C between an unpaired fob and a paired fob occurs when pairing an unpaired fob to the car.
- Interaction D represents the communication between host tools, fobs, and the car through which encrypted packages are transmitted.

9.1 Network Protocol Diagram

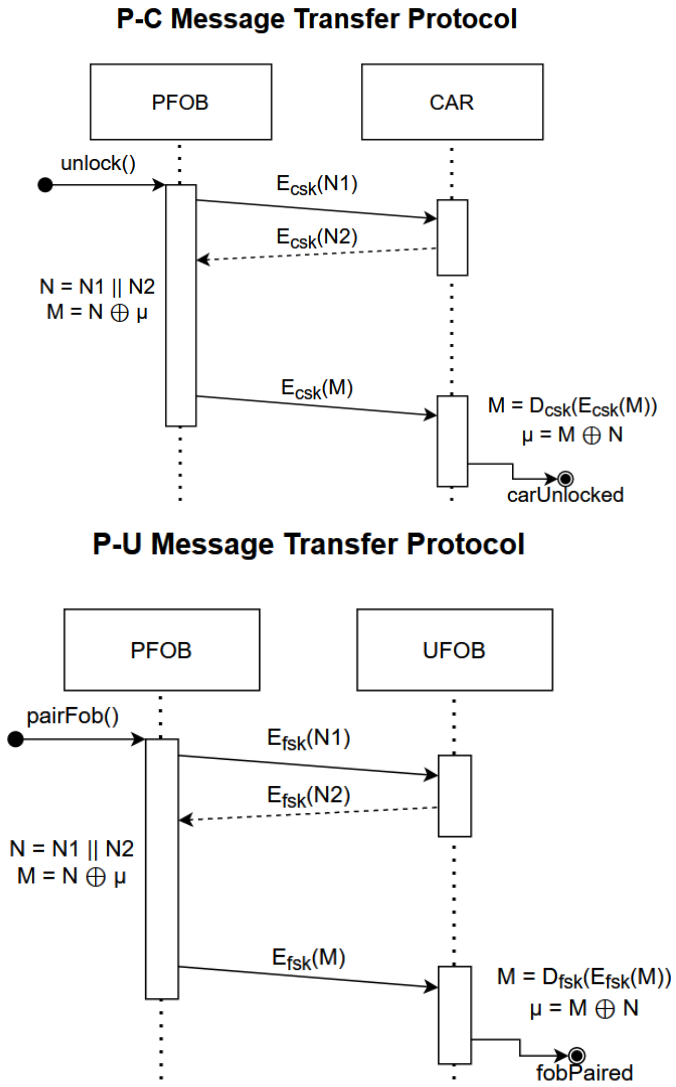

Figure 3

Figure 3 shows the message transfer protocol with the following legends.

- **csk** – car-shared key

- **fsk –** fob-shared key
- **E_x() –** where E() defines an encryption function and x denotes the function key (fsk or csk)
- **D_x() –** where D() defines a decryption function and x denotes the function key (fsk or csk)
- **N** – nonce
- **M –** The transferred message block
- **μ –** denotes the payload
- **| | –** concatenation operator

9.2 Message Packet Diagram

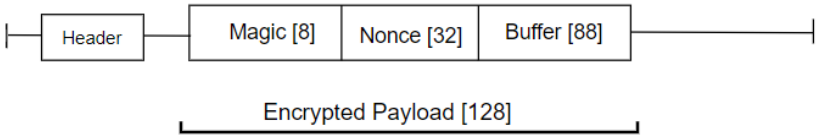

Figure 4

Figure 5 depicts the structure of the message packets. Values in [] denotes the size of each field in bits.

9.3 State Machine Diagram for Pair Fob Feature

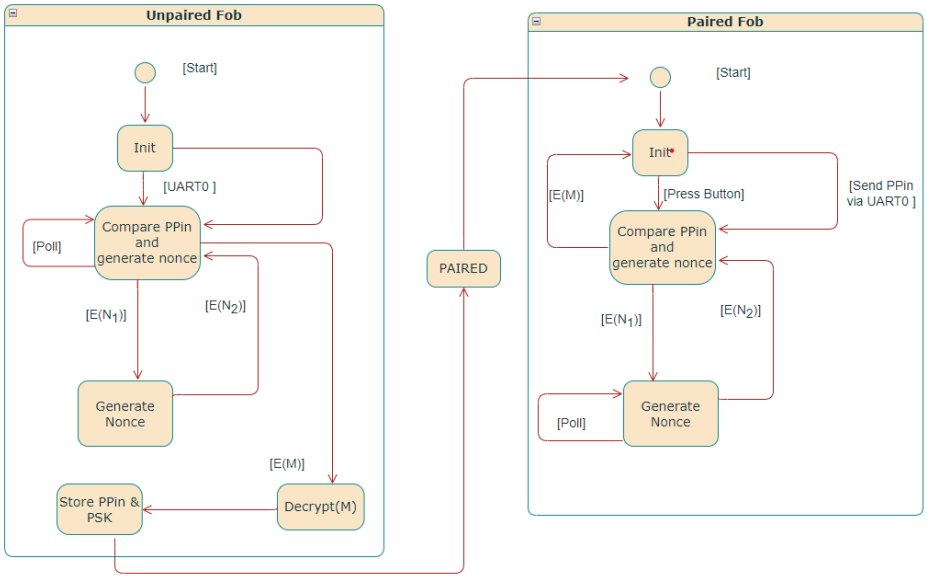

Figure 5

Figure 4 illustrates the state diagram of Pair Fob Feature in Host tool using paired and unpaired fob.4

10 Assumptions

AS1. The unpaired fob is the same for all cars.
AS2. The PPIN (Pairing PIN) is unique to the car.
AS3. The CAR.ID (CAR ID) is unique to the car. It is like VIN#
AS4. The PPIN (Pairing PIN) and CAR.ID (Car ID) are inputs to the host tools.
AS5. The HST.SEC (Host Secrets) will be the implementation secrets for the encryption related items integration into the specific host tool.
AS6. "Authentic" and "Paired" are adjectives used for the fobs received from the manufacturer and has the binaries of the paired fob for a specific Car only. Both these adjectives have been used throughout the document. They both have the same technical meaning.
AS7. With THR4, we believe the attacker cannot mount successful side-channel attacks.
AS8. “New” == “Self-created/malicious,” “Unauthorized” == “Not authorized to use”
AS9. Car Owner Host Tools has associated CAR ID for the Car owned.
AS10. Attacker does not have Host Tools access.

11 Traceability Matrix

Security Requirements Traceability Matrix

Threats	SR#	Requirement Description	Components#	Implementation Status	Test #s
THR1, THR5, THR6	SR1.1	The system shall FR3.4 (Unlock and Start Car) only by P.FOB (paired fob).	PFOB, CAR	Complete	T2.3, T2.4
THR2, THR6	SR2.1	The System shall prevent reading of the P.FOB (paired fob). This includes FOB.FW (fob firmware binary) and FOB.ROM (fob EEPROM Data)	PFOB, FOB, FOB.ROM	Complete	T2.1
	SR3.1	The P.FOB (Paired Fob) shall not leak the P.PIN (Pairing PIN). Constrains FR1 and FR3.3.	PFOB, PPIN	Complete	T2.1
	SR3.2	Altering the signal shall not allow FR3.4 (access to unlock the car).	CAR	Complete	T2.1
THR3, THR4	SR3.3	Replaying the signal shall not allow FR3.4 (access to unlock the car).	CAR	Complete	T2.1
THR1, THR5, THR6	SR4.1	The system shall provision P.FOB (paired fob) and P.PIN (pairing PIN) uniquely for each CAR.FW (Car).	PFOB, PPIN, CAR	Complete	T2
	SR5.1	No one other than the manufacturer shall package new features to a fob.	HT.PF	Complete	T2
	SR5.2	The system shall not allow tampering with the packaged feature	HT.PF	Complete	T2
	SR5.3	The system shall prevent an owner from adding an unpackaged feature.	HT.PF	Complete	T2
THR7	SR5.4	The system shall prevent an attacker from escalating privileges on a feature-limited fob.	HT.PFOB	Complete	T2
THR8	SR6.1	When a car owner paid for an upgraded feature and received an upgrade from the manufacturer then the Car owner shall not be allowed to upgrade it on the other car. (In a situation, car owner has multiple cars from same	HT.EF	Complete	T2

		manufacturer and request and pay for the feature on only one CAR.ID)			
	SR6.2	An attacker shall not be allowed to use feature upgrade when he has paid for an upgraded feature and the manufacturer sends them using other car details	HT.EF	Complete	T2

12 Change Log

Date	Details	Section Ref. link	Version
02/07/2023	Inserted Assets section 3.1	Assets	1.1
02/08/2023	Added SR5.2. to SR5.4., SR6.2. to SR6.4. as per clarification from the organizers (https://mitre-ectf.slack.com/archives/C01G6KVCNHG/p1675859570635999)	**Error! Reference source not found. Error! Reference source not found.**	1.1
02/08/2023	SR4 modification is pending https://mitre-ectf.slack.com/archives/C01G6KVCNHG/p1675859974640469		1.1
02/08/2023	Deleted SR1.1 and SR1.2 as per review comments. The SR1.3 is relabeled to SR1.1	SG1:	1.1
02/08/2023	Deleted SR2.1 as per review comments. The SR2.2 is relabeled to SR2.1	SG2:	1.1
02/08/2023	Changed SR3.2	SR3.2	1.1
02/08/2023	Deleted SR4.1, SR4.3 and SR4.4 as per review comments. The SR4.2 is relabeled to SR4.1 and reworded	SG4: SR4.1	1.1
02/08/2023	Modified SG5 as per review comments. Added SR5.2, SR5.3 and SR5.4 as per Rules V1.1 and clarification. Refer to **Error! Reference source not found.**	**Error! Reference source not found. Error! Reference**	1.1

		source not found. Error! Reference source not found. Error! Reference source not found.	
02/08/2023	Deleted SR6.1 and SR6.2 and added new requirements as per Rules V1.1 and clarification. Refer to **Error! Reference source not found.**	SG6: SR6.1 SR6.2	1.1
2/08/2023	Reworked Threat descriptions using Action->Asset->Harm from Asset->Action->Harm	Threat Identification	1.1
2/09/2023	Modified All SG for the last 2 lines. And SG1 with Justification	SG1	1.1
2/10/2023	Added Assumptions AS6-10	Assumptions	1.1
2/13/2023	Changed the Asset label from HT_PFOB to PFOB and HT_UFOB to UFOB		
2/16/2023	In FR5, FOB.F# and CAR.ID length is changed to 8 bits as per Appendix A-date 02/13/23		
2/18/2023	Added section 7 Implementation Design		
2/23/2023	Added time requirements to Nonfunctional requirements and testing design.	S6, S7	1.1
2/28/2023	Traceability Matrix addition	Section 11	
3/2/2023	Test Design update and tools addition	Section 7	
3/3/2023	Network Protocol Diagram, State machine diagram for Pair Fob	Section 9	

13 Appendix A: Clarifications from Organizers in Slack

Date	Details
02/08/23	From what I understand, the fob has 3 features: Locking/Unlocking the car Feature 1 Feature 2 Regarding SR5: When paring a new fob using an existing fob + pin, how do we decide which features are available to the new fob? Is it decided by the hardware of the second fob? Or do we need to allow the use to choose what feature to grant the secondary fob? Regarding SR6:

	Are the packaged features car-specific or fob-specific? If Fob A and Car A have Feature 1: Can I pair Fob B with Car A and access Feature 1 on Car A on Fob B? Similarly, can I pair Fob A with Car B, losing access to Feature 1 on Car A and gaining access to Feature 1 on Car B?
02/08/23	Unlocking isn't really a feature (it doesn't need to be packaged and enabled on the fob). instead, it is a built-in capability of a fob when it is paired to a car. When pairing a new fob, that fob must only be able to unlock the car it was paired for - it doesn't have to carry over any features enabled on the other fob, and if you want features you should have to enable them explicitly by running the enable feature host tool. features are packaged for a specific car, not a specific fob
02/08/23	SR4: Having an unpaired fob should not allow an attacker to unlock a car without a corresponding paired fob and pairing PIN Without both the paired fob and associated PIN, an attacker should not be able to pair a new fob with the car. Otherwise, an attacker with just the fob or just the PIN could make their own fob. Additionally, an attacker with access to the fob should not be able to determine the PIN, as they would then have both components to properly pair a second fob. After pairing an unpaired fob, can I use the newly paired fob to pair another fob? Eg: Fob A is paired with the car Using Fob A + pairing pin, pair Fob B Using Fob B + pairing pin (no access to Fob A), can I pair Fob C? After pairing an unpaired fob, can the pairing be revoked by the owner? Eg: Fob A is paired with the car Using Fob A + pairing pin, pair Fob B At a later time, can the permissions of Fob B be revoked? If so, what is needed to revoke the permission? (Access to the car, access to Fob A, access to Fob B, pairing pin, or some combination the four) Yes, newly paired fobs should be able to continue pairing new fobs. Once a fob is paired, we aren't requiring your system to be able to unpair it.
02/13/23	We are making a slight change to the size requirements of Car ID and Feature Numbers. Both were previously listed as 32b unsigned integers in the Technical Specifications document. We had teams point out a bug in the example design's handling of larger values in this range. Therefore, we are changing the requirement for both 8b unsigned integers. Your design may still support values in the full 32b range, but you are now only required to support values 0-255 for Car ID and Feature Number.