Medical Infrastructure Supply Chain (MISC) Design

Team RGB
UCCS

April 11, 2024

Contents

1 Purpose

1. Design of secure firmware for Medical Infrastructure Supply Chain (MISC).
2. Design for the protection of global secrets and attestation date from unauthorized users.
3. Test the MISC design for the security goals implementation.
4. Design of the Build Environment and the Build Tools to achieve the security goals.

2 Security Goals Design

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

SG1. The Application Processor should only boot if all the expected chips found on the device are connected and in a fully functional state.

To achieve this goal, SR1 and SR2 are the constraints and will impact FR3.1 and FR3.4.

SG2. The chips found on the device should only boot after the Application Processor has told it to do so and has verified that nothing has been changed about the device.

To achieve this goal, SR3 and SR4 are the constraints that will impact FR3.1 and FR3.4.

SG3. The Attestation Pin and the Replacement token should not be readily available for unauthorized individuals.

SR5 and SR6 are the constraints.

SG4. The attestation data should be kept confidential.

To achieve this goal, SR7 needs to be implemented.

SG5. The integrity and authenticity of messages sent and received in the post-boot secure communications functionality should be ensured.

To achieve this goal, SR8 and SR9 are the constraints and will impact FR3.5.

3 Scope

3.1 Functional Requirements

The functional requirements (labeled as FRx – x corresponds to numbers 1, 2, ... 1.1, 1.2. . . .) The final implementation shall not break them.

FR1. Build MISC System: The Components are built at the secure component facility while the AP is built at the medical device manufacturer's facility, where it is assembled into a final device. It provides the following deliverables:

FR1.1. Build Environment: All project dependencies must be installed utilizing Nix.

FR1.2. Build Deployment: Create a deployment that represents an entire fabrication run of components created by the component manufacturer invoked by the Build Deployment Host Tools.

FR1.2.1. Global Secrets: These must be made read-only, attackers never have access, and the AP and Component Phases, must NOT add new secrets.

FR1.2.2. Deployment: During the attack phase, one Deployment will be built for each scenario.

FR1.3. Build Application Processors and Components: Application Processors and Components will both be built using the relevant Host Tool in any order.

FR2. Create a Medical Device: Firmware for AP and two Components must be completed before this step. Once the previous step is completed, flash

firmware, insert boards into the custom carrier, and connect to the Host Computer.

FR3. MISC Functional Requirements: After the Medical Device is assembled, we must be ready to accept Commands from the Host Tools to the Application Processor. It provides the following deliverables:

FR3.1. List Components: The MISC must be able to list the Component IDs of the Components currently installed on the Medical Device.

FR3.2. Attest: MISC must allow an authorized user to retrieve the Attestation Data that was stored on the Components during the build process only if the user can provide a valid Attestation PIN.

FR3.3. Replace: If a Component(s) fails, the MISC should allow an authorized user to replace it with a new, valid component only if the user can provide a valid replacement token.

FR3.4. Boot: Ensure the integrity of the device and the Components on it, if this check fails, the boot process will be aborted. Otherwise, the MISC prints a boot message, and hands off control of the AP and Controllers to the software that will run the device.

FR3.5. Secure Send and Receive: Must provide a secure communications channel for the AP and Components to use.

FR4. The table below details the host messages that must be used.

Level	Format	Use
Error	%error: message%	Notify of an error/failure. Exits Host Tool
Success	%success: message%	Notify of a successful completion of a function. Exits Host Tool
Info	%info: message%	Provide functionality-relevant information
Debug	%debug: message%	Provide debug information (ignored by testing framework)
Ack	%ack%	Acknnowledge the receipt of a message, requesting more data

FR5. The table below details what messages must be used to communicate if components are connected to the system.

Level	Message Format	Example
Info	P>{Provisioned Component ID prefixed by 0x} \n	P>0x02 \n
Info	F>{Found Component ID prefixed by 0x} \n	F>0x02 \n
Success	List \n	List \n
Debug	%debug: message%	Provide debug information (ignored by testing framework)
Error	Any error message	Internal error \n

FR6. The table below details what boot messages must be printed before booting.

Level	Message format	Example
Info	{Component ID prefixed by 0x}>{Component boot message} \n	0x02>Hello world from a component \n
Info	AP>{AP boot message} \n	AP>Hello world from the AP \n
Success	Boot \n	Boot \n
Error	Any error message	Boot failed \n

FR7. Table 3.4 details the messages that must be printed when replacing a component.

Level	Message format	Example
Success	Replace \n	Replace \n
Error	Any error message	Replace failed \n

FR8. The return attestation function must print all fields of the component's attestation data. This function can make use of the Attestation PIN. The requirements for messages can be seen in the following table.

Level	Message format	Example
Info	C>{Component ID prefixed by 0x} \n	C>0x02
Info	LOC>{Attestation location} \n	LOC>Boston \n
Info	DATE>{Attestation date} \n	DATE>01/01/1970 \n
Info	CUST>Attestation customer \n	CUST>MITRE \n
Success	Attest \n	Attest \n
Error	Any error message	Attest failed \n

3.2 Security Requirements

The implementation of security requirements (SRx – x corresponds to numbers 1, 2, ... 1.1, 1.2....) mentioned below:

SR1. The Application Processor should only boot if all of the expected chips found on the device including sensors that take measurements and actuators are connected.

SR1.1. The FR3.4. (Boot) should only work if the Application Processor and other components are connected.

SR2. The Application Processor Should only boot if all of the expected chips found on the device including sensors that take measurements and actuators are in a fully functional state.

SR2.1. The FR3.4. (Boot) should only work if the Application Processor and other components are in a fully functional state.

SR3. The chips found on the device including sensors that take measurements and actuators should only boot after the Application Processor has told it to do so.

SR3.1. After SR2.1., FR3.4. (Boot) of chips found on the device including sensors that take measurements and actuators.

SR4. The chips found on the device including sensors that take measurements and actuators should only boot after the Application Processor has verified that nothing that should not be changed has changed about the device.

SR4.1. The FR2.3. (Attest) should be used by the Application Processor to determine if anything has changed about the components.

SR5. The Attestation Pin should not be readily available for unauthorized individuals.

SR5.1. The Attestation Pin may be accessible indirectly by the attacker in some scenarios.

SR6. The Replacement Token should not be readily available for unauthorized individuals.

SR6.1. The Replacement Token may be accessible indirectly by the attacker in some scenarios.

SR7. The component attestation data should be kept confidential.

SR7.1. The FR3.2. (Attest) should keep the attestation data confidential unless the correct Attestation PIN is used.

SR8. The integrity of messages sent and received in the post-boot MISC secure communications functionality should be ensured.

SR8.1. For FR3.5. (Secure Send and Receive), messages should be secured so that unauthorized individuals are not able to alter the messages.

SR9. The authenticity of messages sent and received in the post-boot MISC secure communications functionality should be ensured.

SR9.1. For FR3.5. (Secure Send and Receive), messages should be secured so only authorized individuals are able to send and receive messages.

3.3 Non-Functional Requirements

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

NFR1. The MISC system shall comply with the timing requirements shown below.

Operation	**Maximum Time for Completion**
Device Wake	1 second
Attestation	3 seconds
List Components	3 seconds
Boot	3 seconds
Replace Component	5 seconds

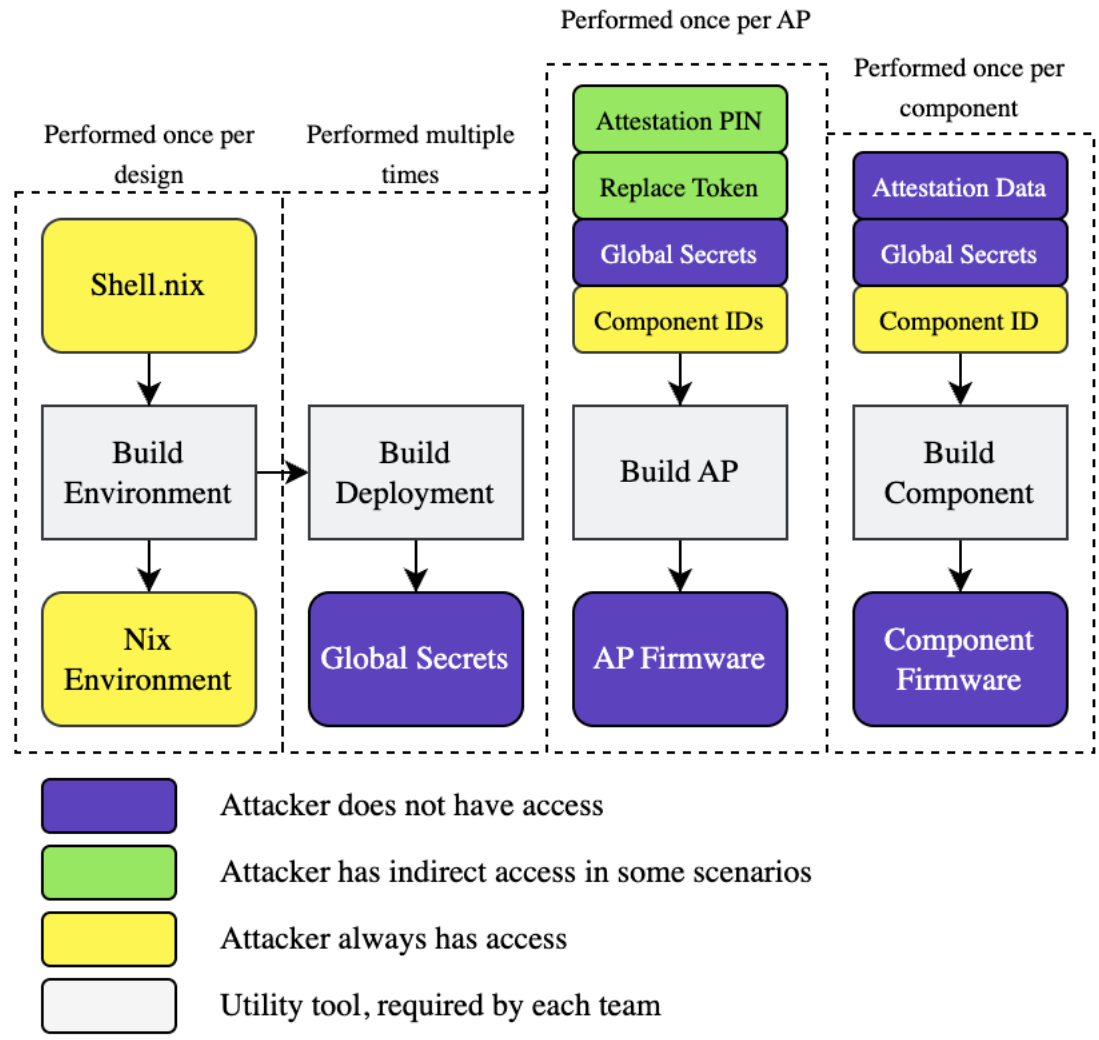
NFR2. The MISC system shall comply with the firmware size requirements shown below.

Component	**Size**
Component ID	4 bytes (Last byte is valid I2C address)
Replacement Token	16 bytes
Attestation Pin	6 bytes
AP Boot Message	Max 64 bytes
Attestation Customer Name	Max 64 bytes
Attestation Manufacture Location	Max 64 bytes
Attestation Date	Max 64 bytes
Component Boot Message	Max 64 bytes

NFR3. The firmware size must stay within the address spaces shown below.

Starting Address	0x10010000
Ending Address	0x10045FFF

4 Build Tools and Environment Design



Mitre ©

5 Test Design

Testing of the functional requirements, security requirements, and non-functional requirements as per *Section 3: Scope*. Testing will be done to ensure that any security requirements do not break functionality except for any constraints in *Section 2: Security Goals*.

T1.1 Static Code Analysis using the GitHub built-in CodeQL tool will ensure there are no new patches, numerical errors, input validation issues, race conditions, path traversal vulnerabilities, as well as checks for pointers and references within the codebase.

T1.1.1 Code scanning will encompass linting the code to ensure there are no insecure coding errors that could potentially serve as easy attack vectors for attackers. This validation will be performed using the IntelliSense Code Linter integrated within Visual Studio Code.

T1.2 Test secure communication between the AP and components. verifying that the communication channel is encrypted, messages are authenticated, and unauthorized access is prevented. (consult with team to find the tools we can use/ Access control testing frameworks like OWASP ZAP or Burp Suite can be used to simulate different user roles and test access control policies).

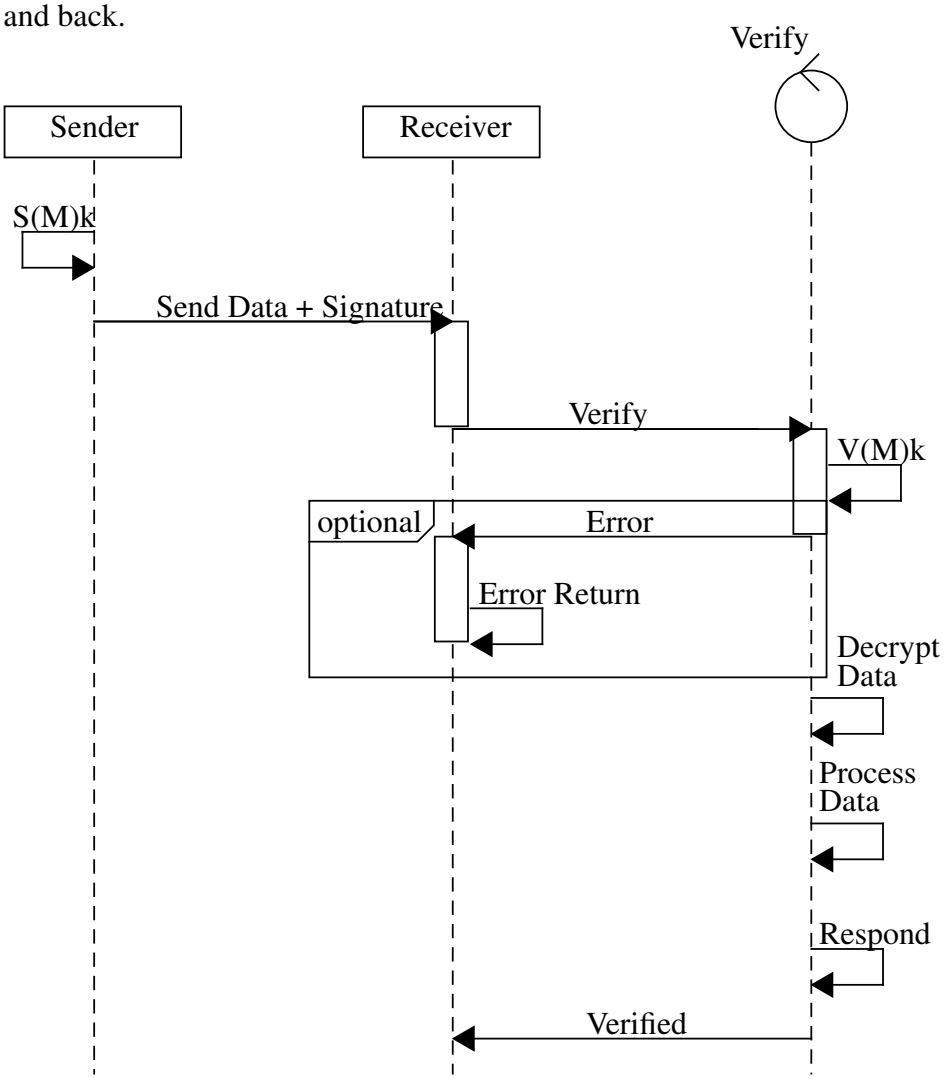
6 Threat Identification and Mapping to Security Requirements

Threat Number	**Threat Descriptions**	**Security Requirement Number**
THR1	An attacker can manipulate the firmware leading to skipped validation checks and booting invalid firmware.	SR1.1., SR2.1.
THR2	An attacker can boot a component without the AP present leading to unauthorized operation of the component.	SR1.1., SR2.1., SR3.1., SR4.1.
THR3	An attacker can boot a component without it being valid leading to the access to components and data.	SR2.1., SR3.1.
THR4	An attacker can boot a component without it being valid leading to potential device malfunction.	SR2.1., SR3.1.
THR5	An attacker can access credentials leading to data breaches.	SR4.1., SR5.1., SR6.1., SR7.1.

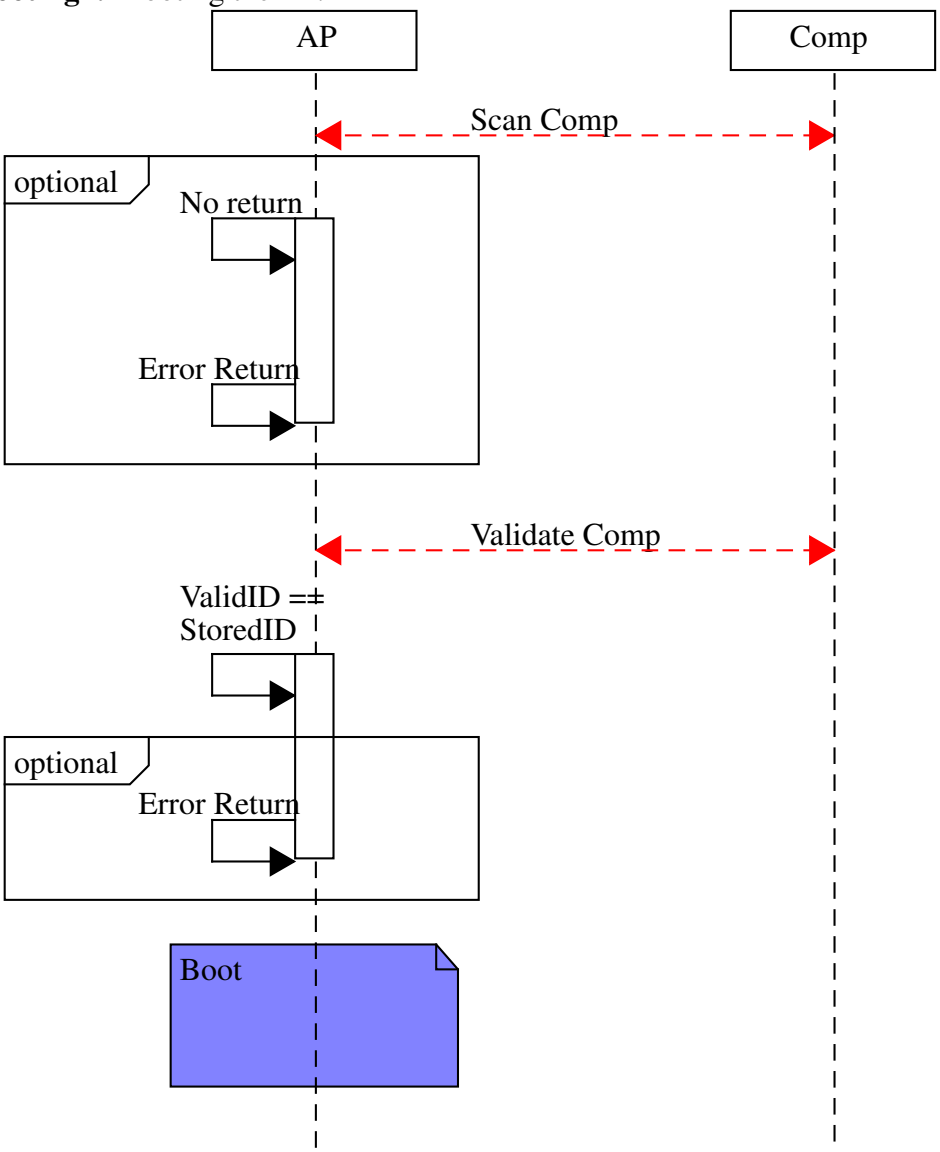
THR6	An attacker can access the Replacement Token leading to unauthorized components compromising functionality.	SR6.1., SR7.1.
THR7	An attacker can access the Attestation PIN or Replacement Token leading to unauthorized firmware updates.	SR5.1., SR6.1., SR7.1.
THR8	An attacker can access credentials leading to device malfunctions.	SR5.1., SR6.1., SR7.1.
THR9	An attacker can intercept messages exchanged between components leading to compromised communications.	SR8.1., SR9.1.
THR10	An attacker can access side-channels leading to sensitive information leaks.	SR8.1., SR9.1.

7 Sequence Diagrams

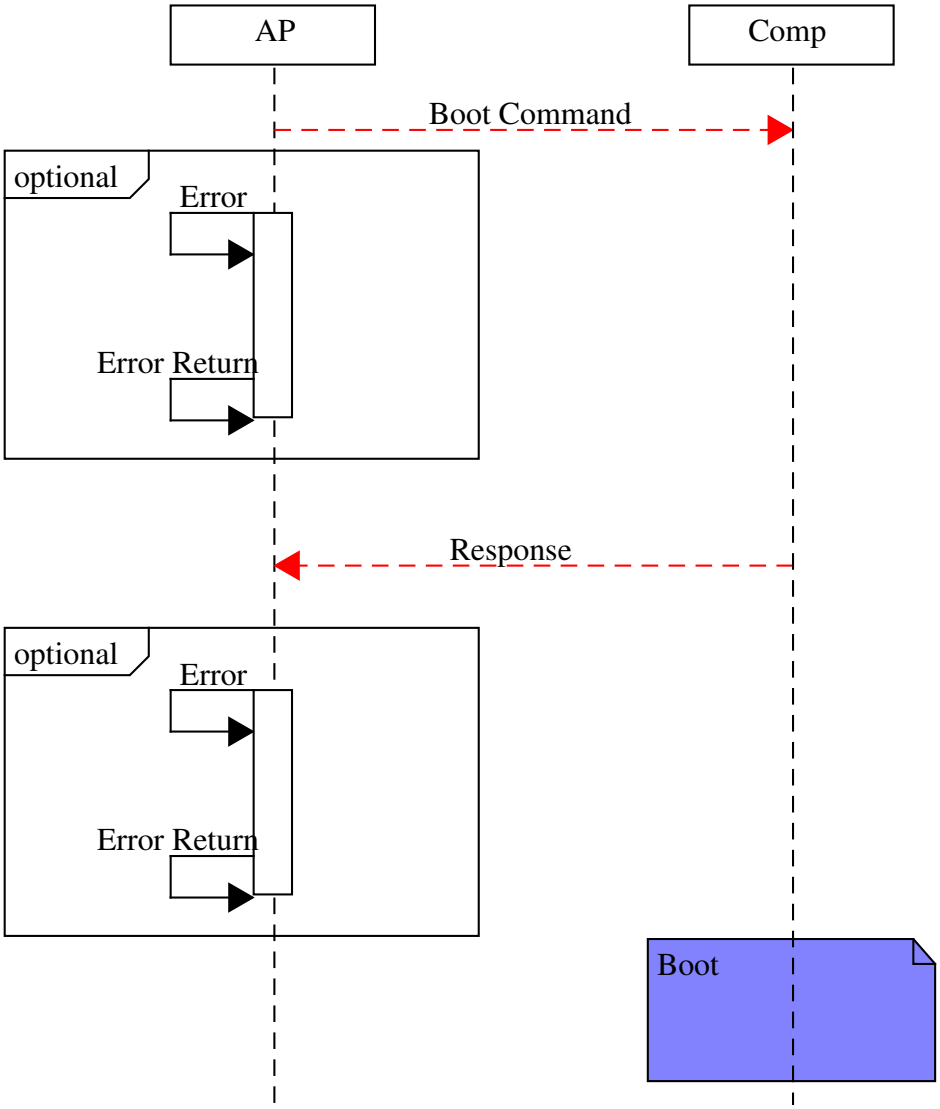
7.1 Messaging Service: Data being sent from the AP to different components and back.



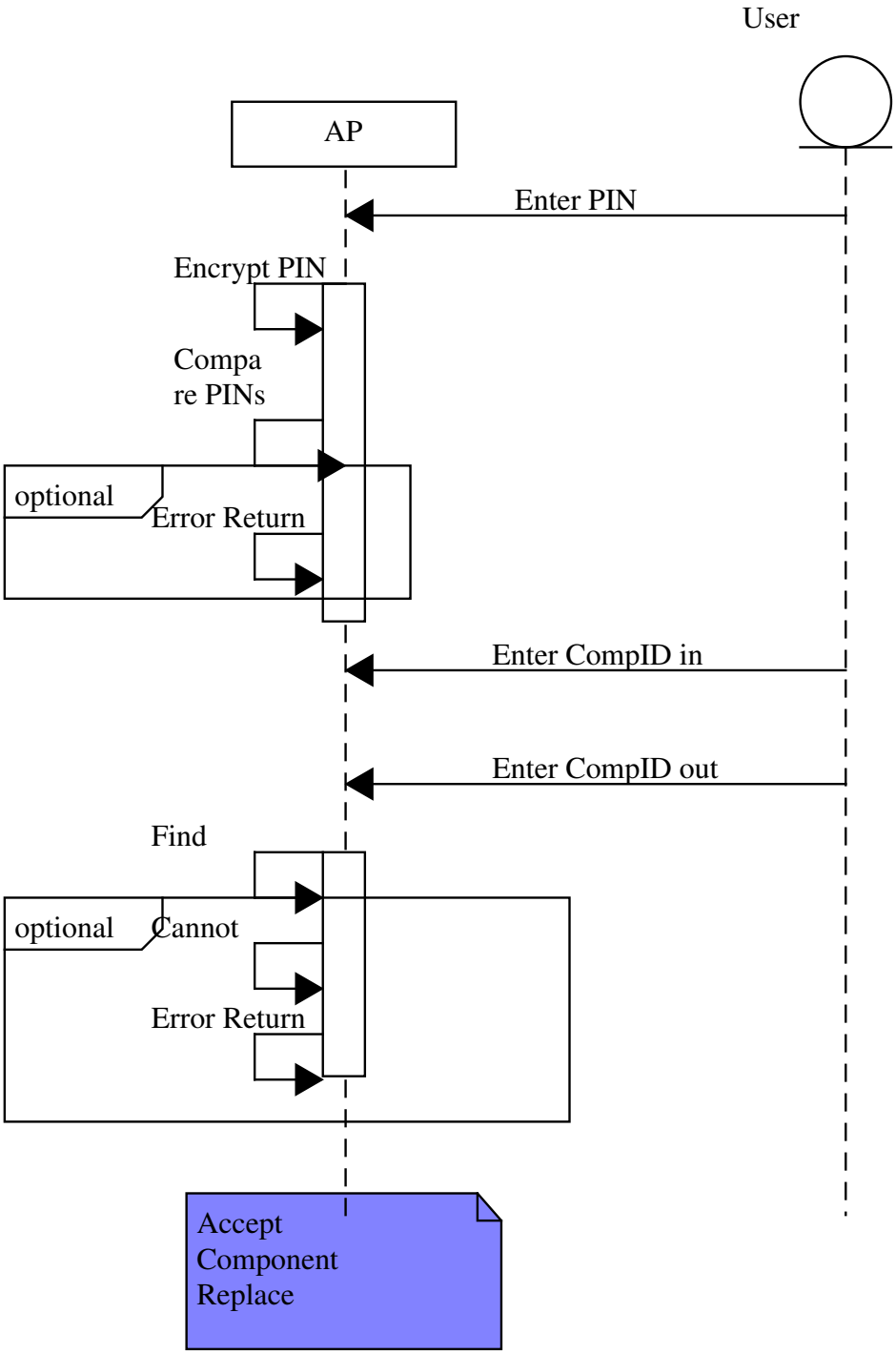
7.2 Booting1: Booting the AP.



7.3 Booting2: Booting the components.



7.4 Component Changing: Changing some pieces of components.


User
AP
Enter PIN
Encrypt PIN
Compa
re PINs
optional
Error Return
Enter CompID in
Enter CompID out
Find
optional
Cannot
Error Return
Accept
Component
Replace

8 Protocols

8.1 TLS For Secure Messaging: TSL will be implemented using a public key and private key using elliptic curve cryptography (ecc) for the keys. The packet that is sent will have a signature added to the data so that it can be verified and then it will return the length of the data if the packet comes from a secure source.

8.2 Encryption: The encryption standard will be AES-256 bit. There will be 265 bit ciphers used for the AP_PIN and the pin used during the boot command. These ciphers are generated using a common randomly generated pin whenever the validate_pin function is called. The pins consist of randomly generated numbers for the key, iv, and salts. The two ciphers are then compared and the pin is validated. The same process will happen for the validate_token.

9 Assumptions

AS1. The Attacker does not have access to Global Secrets, Attestation Data, AP Firmware, Component Firmware, or the Component Replaced Message.

AS2. A valid attestation PIN and Token are required for startup on the device.

AS3. AP is responsible for ensuring the integrity of the device.

10 Tractability Matrix

Threats	SR #	Requirement Description	Implementation Status	Test #
THR1 THR2	SR1.1.	The FR3.4. (Boot) should only work if the Application Processor and other components are connected.		T1
THR1 THR2 THR3 THR4	SR2.1.	The FR3.4. (Boot) should only work if the Application Processor and other components are in a fully functional state.		T1
THR2 THR3 THR4	SR3.1.	After SR2.1., FR3.4. (Boot) of chips found on the device including sensors that take measurements and actuators.		T2
THR2 THR5	SR4.1.	The FR2.3. (Attest) should be used by the Application Processor to determine if anything has changed about the components.		T2
THR5 THR7 THR8	SR5.1.	The Attestation Pin may be accessible indirectly by the attacker in some scenarios.		T3

THR5 THR6 THR7 THR8	SR6.1.	The Replacement Token may be accessible indirectly by the attacker in some scenarios.		T4
	SR7.1.	The FR3.2. (Attest) should keep the attestation data confidential unless the correct Attestation PIN is used.		T5
THR9 THR10	SR8.1.	For FR3.5. (Secure Send and Receive), messages should be secured so that unauthorized individuals are not able to alter the messages	In Progress	T6
	SR9.1.	For FR3.5. (Secure Send and Receive), messages should be secured so only authorized individuals are able to send and receive messages	In Progress	T6