Medical Infrastructure Supply Chain (MISC) System Design

MITRE eCTF 2024
Team **Cacti**
University at Buffalo

1 Introduction

The Medical Infrastructure Supply Chain (MISC) system simulates a supply chain security solution for microcontrollers on a medical device. Our design consists of three parts: host computer, Application Processor (AP), and Component (CP). The MISC system includes one AP and two CPs connected via an I2C bus. The host computer is a general-purpose computer communicating with the AP over a serial interface. The host tools in the host computer send commands to and receive messages from the AP.

The following summarizes the features of the MISC system:

- The AP can list provisioned and presented CP IDs.
- The whole system will only boot up when all the provisioned CPs are present and valid.
- The AP can retrieve attestation data from a CP with the correct PIN code.
- After replacing a CP, the AP needs to be notified of the change of a provisioned CP ID.
- The integrity and authenticity of communication messages between AP and CP are protected.

2 Security Requirements

This section defines the security requirements of our design.

The AP and CPs must be valid (built by the organizers) for the MISC system to work properly.

2.1 SR1

The Application Processor (AP) should only boot if all expected CPs are present and valid.

How we address it: The AP needs to verify all the present CPs have provisioned IDs and that they are valid CPs built by the organizers. We use the public-key cryptography algorithm. AP holds the public key, while CP holds the private key. When the host computer sends the boot command to the AP, the AP generates a challenge using a random number and sends the validation command along with the challenge to one specific CP. The CP encrypts the challenge along with its ID using the private key and sends the decrypted data back to the AP. The AP then decrypts the received data by using the public key to check if the decrypted random number matches the originally generated one and if the ID matches the corresponding CP ID. If they match, verification succeeds. The AP could tell this specific CP is valid. The AP does this process for all the presented CPs.

2.2 SR2

CPs should only boot after being commanded to by a valid AP that has confirmed the integrity of the device.

How we address it: A CP needs to verify if the boot command is given by the valid AP. We use the RSA algorithm. AP holds the private key, while CP holds the public key. First, the AP sends the boot command to one specific CP. Then, this CP generates a random number and sends it number to the AP as the challenge. The AP encrypts the challenge along with the CP ID as the response and sends it back to the CP. The CP decrypts the response, if the decrypted number matches the originally generated random number and the CP ID is correct, validation passes and the CP boots up.

2.3 SR3

The Attestation PIN and Replacement Token should be kept confidential.

How we address it: The Attestation PIN and Replacement Token will not be stored in plain text or hard-coded in the program (e.g., use the macro generated at the build phase). The PIN and the Token will be hashed using a keyed-hash algorithm, and the two hash strings will be stored in the flash memory when the AP board boots for the first time. Both the AP and the CP hold the same hash key. After receiving a PIN or a token, the AP will apply SHA1 to the input, and compare the hash string with the saved one. To prevent a brute force attack, the AP will delay randomly up to 5 seconds after receiving a wrong PIN or token.

2.4 SR4

CP Attestation Data should be kept confidential.

How we address it: Use AES-128 to encrypt the CP Attestation Data and store it in the flash memory when the board boots for the first time. To retrieve the data, a CP uses the same key to decrypt. To write the plain CP Attestation Data to the I2C buffer, a CP needs to verify if the request message is from the valid AP. RSA algorithm and challenge-response mechanism are used for the verification process. The AP holds the private key, and the CP holds the public key. The AP sends a request message to the CP The CP generates a random number and sends it to the AP as the challenge. The AP encrypts the challenge along with the CP ID using the private key and sends it back to the CP as the response. The CP decrypts the response using the public key and checks if the decrypted random number and ID match. Verification succeeds if the received plain number matches the generated one.

2.5 SR5

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

How we address it: For integrity, all communications in the I2C bus have the data checksum. The checksum employs a keyed hash algorithm for the data. The key is generated from the Global Secret to ensure both the AP and CP have the same key value. After a CP receives a message or the AP reads a message, the message will always be validated using checksum. For authenticity, a CP needs to validate if the sender is the valid AP before receiving the message, and the AP needs to validate if a message is read from a valid

CP before accepting the read message. We use the RSA algorithm with two key pairs and the challenge-response mechanism. For the first key pair, AP holds the private key, while CP holds the public key. For the second key pair, AP holds the public key, while the CP holds the private key. For the scenario of the AP sending a message to a CP, the AP first sends the request message to one specific CP. The CP sends a random number as the challenge number to the AP, and the AP encrypts the received number along with the CP ID as the response. The AP then sends the transmitted data and response back to the CP. The CP decrypts the response to verify if the decrypted number matches the original generated one and if the ID is correct. The CP receives this message if validation passes. For the scenario of AP reading a message from a CP, the AP sends a random number as a challenge to the specific CP. The CP encrypts the number along with the ID, then puts the message and decrypted number into the I2C buffer waiting for the AP to read. The AP reads the data from the CP, and decrypts to check if the number matches the original one and if the ID is correct. The AP will receive the message if validation passes.

3 Security Implementations

3.1 Build MISC System

3.1.1 Build Deployment

Attackers will NEVER be given access to the Deployment Secrets generated in this step.

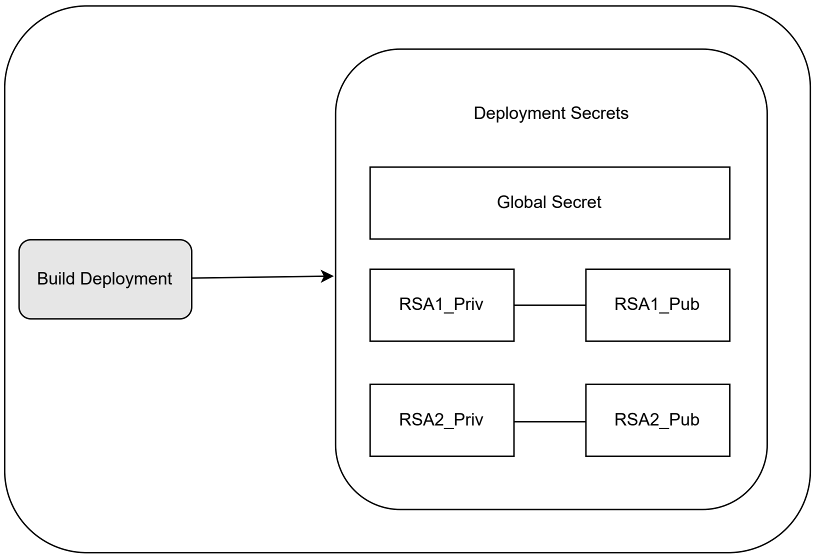

Figure 1: Listing Sequence

Figure 1 shows the Deployment Secrets. During the building deployment process, a Global Secret and two pairs of RSA public and private keys will be generated. The Global Secret will be shared among the AP and all the CPs. AP will store the private key of `RSA1` and the public key of `RSA2`. CP will store the private key of `RSA2` and the public key of `RSA1`. The `RSA1` keys are for the AP to validate a CP, while `RSA2` keys are for a CP to validate the AP. We generate the Deployment Secrets randomly when building deployment to prevent the attacker from retrieving it through the source code.

3.1.2 Build AP and CP Firmware

To build the AP firmware, the number of provisioned CPs, provisioned CP IDs, attestation PIN code, and replace Token need to be provided.

To build the CP firmware, the CP ID and attestation data need to be provided. The attestation data includes location, date, and customer name.

3.2 Load Devices and Booting

3.2.1 Load AP and CP Firmware

The firmware will be loaded by the provided host tool, teams are not allowed to modify this step.

3.2.2 First Booting

Devices will be initialized at the first boot. The AP will store provisioned CP IDs in plain text, the private key of `RSA1`, and the public key of `RSA2` in the flash memory. A hash key is generated based on the Global Secret for the keyed-hash algorithm. The PIN and replace Token will be keyed-hashed, and the AP stores the hash values in the flash. The CP stores the private key of `RSA2` and the public key of `RSA1`. An AES-128 key will be generated based on the Global Secret to encrypt the attestation data. The encrypted data will be stored in the flash.

3.3 Functionalities

3.3.1 Listing

The Listing functionality checks the provisioned CP IDs of the APs and the IDs of all the presented CPs. As CP IDs are not secrets, there is nothing to protect during this process.

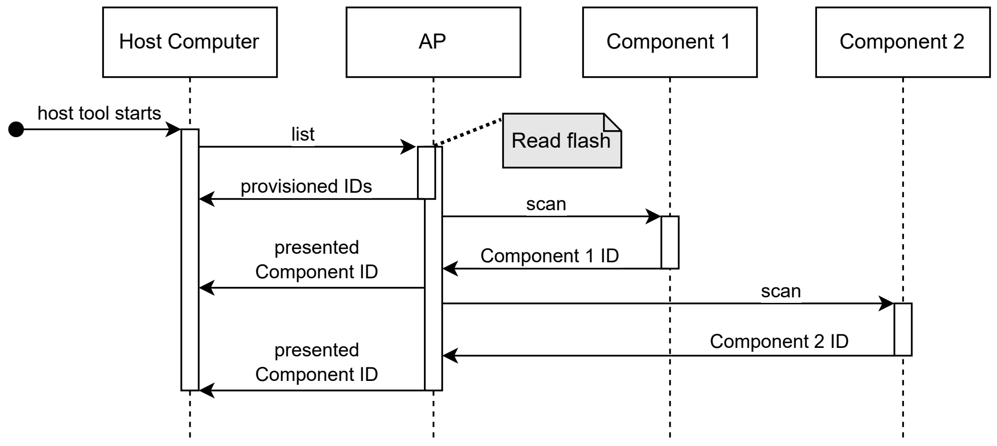

Figure 2: Listing Sequence

The process (shown in Figure 2) is described as follow:

1. The host computer sends the `list` command through USB to the AP.

2. The AP reads all the provisioned CP IDs from its flash memory and sends the IDs back to the host computer.

3. The AP sends the `scan` command to all the possible CP IDs one by one over the I2C bus.

4. For one specific ID, if the CP with this ID is presented, the CP puts the ID value in the I2C buffer for the AP to read.

5. The AP reads the ID value from this CP and sends the ID value to the host computer as one present CP ID.

3.3.2 Replace

The Replace functionality is for replacing a CP on the medical device. The technician in a repair station needs to tell the AP which CP ID is no longer provisioned and what the new CP ID with supplying the correct replacement Token. The replace Token will be hashed before comparing with the saved Token hash value.

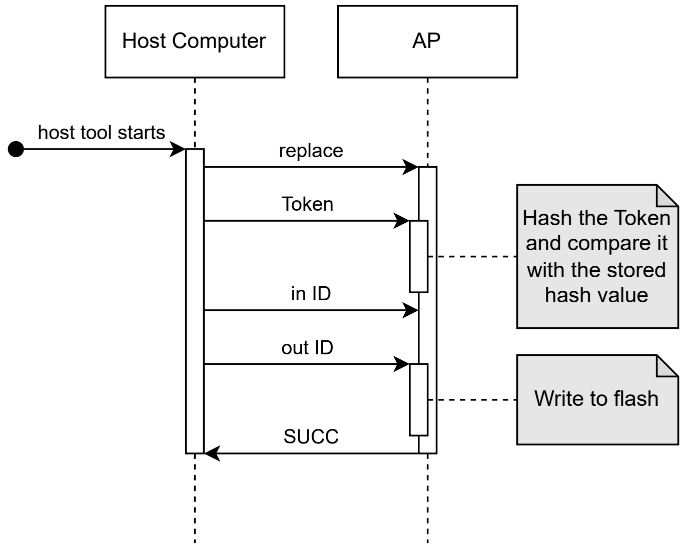

Figure 3: Replace Sequence

The process (shown in Figure 3) is described as follow:

1. The host computer sends the `replace` command through USB to the AP.

2. The AP enters the replace mode and waits for the host computer to send the replace Token.

3. After receiving the Token, the AP applies the keyed-hash algorithm to the Token and compares it with the stored hash value. All the characters in the hash value are compared one by one. The comparison time is always constant no matter if the received Token is correct or not. If a wrong Token is received, the AP will randomly delay for up to 5 seconds to mitigate brute force and then terminate the replace mode. For a correct Token received, the AP will wait for the host computer to send the provisioned and new IDs.

4. The AP modifies the flash memory after receiving the two IDs and sends a success message to the host computer.

3.3.3 Attestation

The functionality gets the attestation data from one specific CP by supplying the correct PIN code and the CP ID. The attestation data is sensitive and encrypted when storing, so decrypting is necessary when retrieving the data from a CP. Also, the PIN code will be hashed before comparing with the saved PIN code hash value.

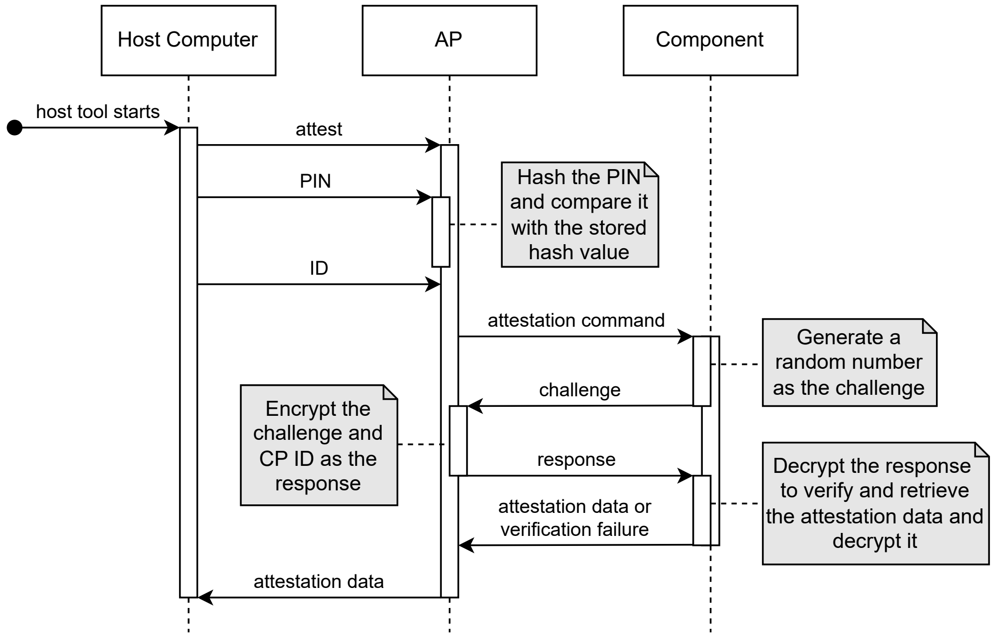
Host Computer
AP
Component
host tool starts
attest
PIN
Hash the PIN and compare it with the stored hash value
ID
attestation command
Generate a random number as the challenge
challenge
Encrypt the challenge and CP ID as the response
response
Decrypt the response to verify and retrieve the attestation data and decrypt it
attestation data or verification failure
attestation data

Figure 4: Attestation Sequence

The process (shown in Figure 4) is described as follow:

1. The host computer sends the attest command to the AP
2. The AP enters the attestation mode and waits for the host computer to send the PIN code
3. After receiving the PIN code, the AP applies the keyed-hash algorithm to the PIN and compares it with the stored one. The comparison time is always constant to matter of the PIN correctness. The AP will randomly delay for up to 5 seconds to mitigate brute force and then terminate the attestation mode for a wrong PIN. When given a correct PIN, the AP tries to acquire the attestation data from the specific CP with the ID given by the host computer.
4. The AP sends the attestation command to the CP.
5. The CP generates a random number and sends it as a challenge to the AP.
6. The AP encrypts the challenge along with the CP ID using the `RSA1_Pri` key and sends it as the response back to the CP. The response is represented as $response = Enc_{RSA1\_Pri}(challenge|ID)$.
7. The CP decrypts the response with the `RSA1_Pub` key to check if the decrypted number matches the originally generated number and if the CP ID is correct. If validation fails, the CP sends the verification failure message to the AP, and the AP randomly delays up to 5 seconds to prevent brute force and terminate the attestation mode. Otherwise, the CP retrieves the attestation data, decrypts it using the AES algorithm, and sends the plain attestation data to the AP.

8. The AP sends the attestation data to the host computer.

3.3.4 Boot

This functionality requires the AP to test if all the provisioned CPs are present and valid before the AP boots. The AP sends commands to all the CPs to let them boot, and the CPs must ensure that the boot command is from the valid AP before booting.

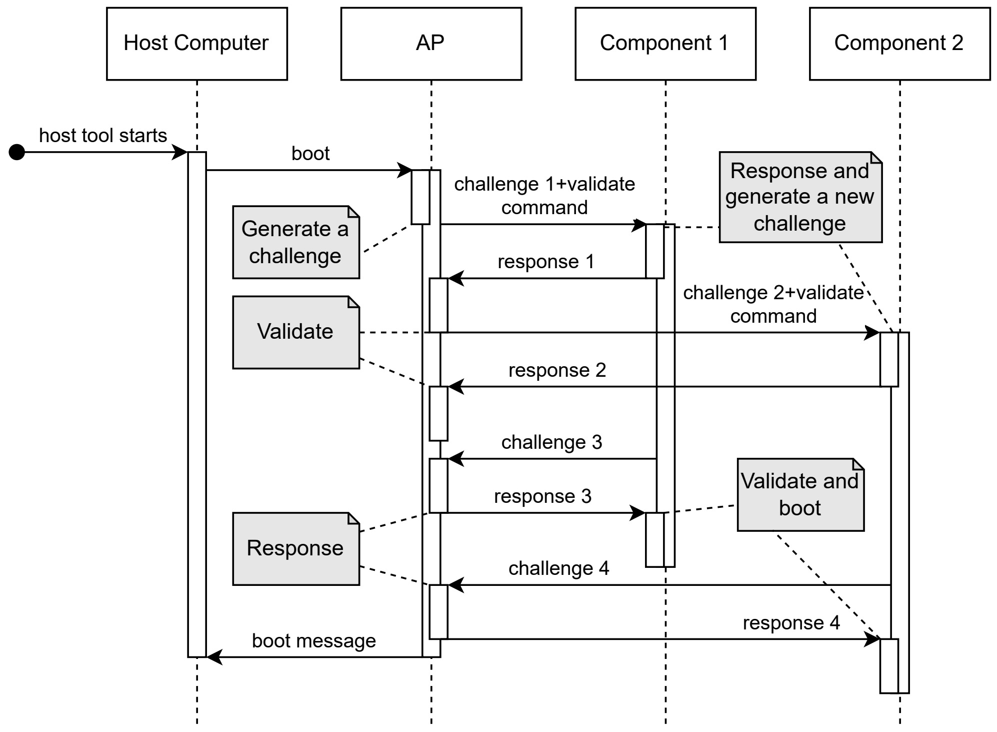

Figure 5: Boot Sequence

The process (shown in Figure 5) is described as follow:

1. The host computer sends the boot command to the AP.

2. The AP validates all the provisioned CPs one by one to check if they are present.

3. For each provisioned CP, the AP generates a random number as the challenge. It then sends the challenge and validation command message to the corresponding CP.

4. After receiving the challenge and validation message, the CP makes up the response by encrypting the challenge along with the CP ID with the `RSA2_Pri` key. The response can be represented as $response = Enc_{RSA2\_Pri}(challenge_{fromAP}|ID)$. It then creates a new challenge by generating a random number awaiting for the AP the read.

5. The AP reads the response from the CP and validates the response by decrypting it to check if the decrypted challenge is the same as the originally generated one and if the CP ID is correct.

6. If any validation fails, the AP will randomly delay for up to 5 seconds to mitigate brute force and terminate the booting process.

7. If all the CPs are present, the AP then tries to boot all the them.

8. For each CP, the AP reads the challenge from the CP and encrypts it along with the CP ID with the `RSA1_Pri` key to make up the response, then sends the response back to the CP. The response is represented as $response = Enc_{RSA1\_Pri}(challenge_{fromCP}|ID)$.

9. The CP validates the response by decrypting it with the `RSA1_Pub` key to check if the decrypted number is the same as the original generated random number and if the decrypted CP ID is correct. The CP boots if validation succeeds.

3.3.5 POST-BOOT Communication

After the AP and CPs boot up, the communication between the AP and a CP is through the I2C bus. The AP is the master device, while the CPs are slave devices. Thus, the AP can send messages to or read messages from a CP. However, the CP cannot send messages directly to the AP in normal situations. The integrity and authenticity of the communication are necessary.

Note : A checksum is used for each sending and reading for communication integrity.

A checksum is calculated by a keyed-hash algorithm with a key generated by the Global Secret. All the data (except for the checksum) in one communication is a packet, which is represented as $Packet = Data + Checksum$. The $Data$ may contain a message, command, challenge, and response. The term message means the real meaningful message the AP intends to send or read. A checksum calculation includes all the data in a communication, which is represented as $Checksum = Hash_K(data)$ (K is the key). After the AP or a CP receives a packet, it will validate the integrity of the packet by using the same keyed-hash algorithm with the same key to calculate the checksum again. If the calculated checksum does not match the received one, attack detected and the system terminates this communication.

The checksum generation and validation will be omitted in the following descriptions of the sending and receiving processes. The descriptions below only focus on validating the AP or a CP for message authenticity.

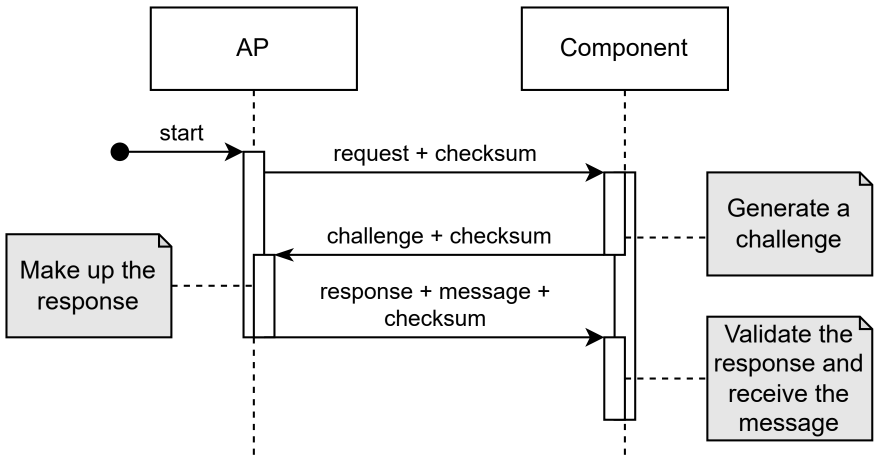

Figure 6: Sending Message Protocol Sequence

The sending message process is shown in Figure 6.

1. The AP sends the sending request command to one specific CP.

2. The CP generates a random as the challenge for the AP.

3. The AP encrypts the challenge along with the CP ID using the `RSA1_Pri` key as the response for the CP. The AP also includes the message along with the response in the packet sent to the CP.

4. The CP validates the response by decrypting it to compare the number with the originally generated number and check if the ID is correct. If validation is successful, the CP receives the message.

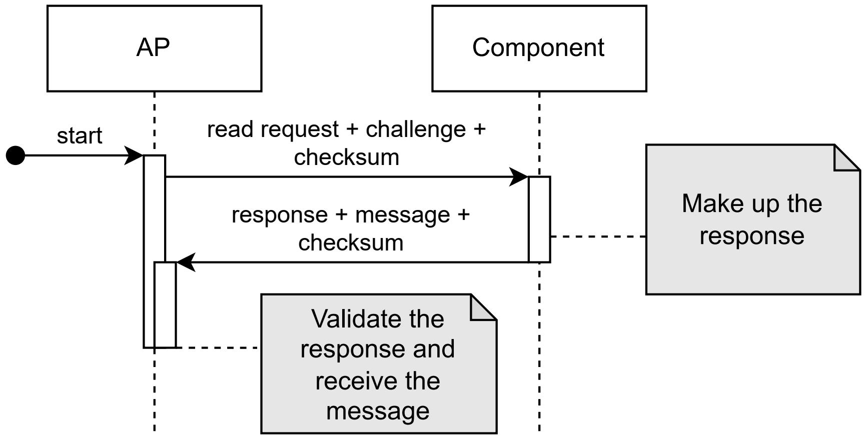
AP
Component
start
read request + challenge + checksum
response + message + checksum
Make up the response
Validate the response and receive the message

Figure 7: Reading Message Protocol Sequence

The reading message process is shown in Figure 7.

1. The AP generates a challenge using a random number and sends the read request command along with the challenge to one specific CP.

2. This CP prepares for the message and makes up the response by encrypting the challenge along with the CP ID with the `RSA2_Pri` key, waiting for the AP to read.

3. The AP read the packet from the CP.

4. The AP decrypts the response using the `RSA2_Pub` key to validate if the decrypted number matches the originally generated random number and if the ID is correct. If validation succeeds, it will receive the message.